

UNIVERSIDAD AUTONOMA DE MADRID

ESCUELA POLITECNICA SUPERIOR



Grado en Ingeniería Informática

TRABAJO FIN DE GRADO

Adaptación de Licitación Electrónica de Pixelware al proyecto europeo "e-SENS eTendering Pilot"

Galo Andrés Lugo Cuji
Tutor: Alberto Chacón Ludeña
Ponente: Francisco Saiz López

MAYO 2017

Adaptación de Licitación Electrónica de Pixelware al proyecto europeo "e-SENS eTendering Pilot"

AUTOR: Galo Andrés Lugo Cuji
TUTOR: Alberto Chacón Ludeña

Pixelware S.A.
Dpto. de Ingeniería Informática
Escuela Politécnica Superior
Universidad Autónoma de Madrid
Mayo de 2017

Resumen

Este Trabajo de Fin de Grado tiene el fin de conectar el entorno de licitación electrónica de la empresa Pixelware con distintos entornos repartidos por Europa mediante una red SMP/SML para la compartición de información acerca de expedientes de contratación entre distintas plataformas de Licitación, accesibles tanto para los Organismos de Contratación como para los licitadores. Para ello se planea diseñar un módulo de intercomunicación basado en mensajes XML transportados mediante la tecnología RESTful que haga de interlocutor.

En específico se trabajará con la compartición y comunicación entre sistemas de Licitación Electrónica en la fase previa a la resolución de ofertas, donde se dan casos de uso como son la expresión de interés por ofertar un expediente externo o la necesidad de recepción de documentos sobre estas ofertas.

Debido a la importancia de una comunicación correcta entre distintos entornos de diversas localizaciones y con información representada de manera variables y en otros lenguajes, se usan documentos estandarizados de representación de datos sobre Licitación Electrónica a nivel europeo (UBL, ESPD, SBD) y encapsulación de la información en un contenedor autenticado (ASIC-e), además de la validación pertinente de tanto el formato como el contenido de los mensajes.

Para el correcto envío, recepción y manejo de los mensajes deberemos crear distintos elementos que se acoplen al entorno de Pixelware y que puedan transformar la información de los distintos documentos obtenidos al modelo de datos del entorno, además de añadir elementos necesarios para el envío de información al modelo de datos estándar.

Para llevar a cabo estas tareas se ha creado un módulo independiente del entorno de Pixelware que hará de nexo entre las distintas implementaciones de Pixelware y los demás entornos europeos. Para ello nos haremos uso de la creación de servidores basados en arquitectura RESTful mediante Spring Boot, la librería *contratación-core* de Pixelware, diseñada para la representación del modelo de datos de Pixelware en entorno Java y algunas librerías auxiliares necesarias, como una librería basada en la creación y el manejo de ficheros ASIC-e o una librería para la validación de los documentos mediante ficheros basados en lenguajes de validación de reglas.

La manera en la que la implementación se ha puesto a prueba las distintas implementaciones es mediante connectathons, una serie de e-meeting de pruebas a nivel europeo con los distintos centros que también trabajan en este proyecto a lo largo del segundo trimestre de 2016 y el primer trimestre de 2017. Estos consisten en la realización de una serie de pasos de intercambio de mensajes en vivo entre los distintos participantes del proyecto y la comprobación de los resultados de los distintos pasos realizados.

Palabras clave

Licitación electrónica, Europa, XML, RESTful, estandarización, UBL, SBD, validación, Pixelware, módulo independiente, Spring Boot, ASIC-e, Schematron, connectathons

Abstract

This bachelor thesis looks for the connection of the Pixelware's electronic tendering environment with various implementations across Europe through an SMP/SML network for sharing information about procurement projects between Contracting Authorities and Economic operators. For making this happen, an intercommunication module based on the sharing of XML messages using RESTful technology must be designed.

We will work with the sharing and communication between Electronic Bidding systems in the phase prior to the resolution of offers, such as the expression of interest in offering an external file or the need to receive documents on these offers.

Due to the importance of a correct communication between different environment at different locations and with information in different representations and languages, the module use standard European data model representations (UBL, ESPD, SBD) and encapsulation of the information in a signed container, in addition to the relevant validation of both the format as the content of the messages.

For the correct sending, receiving and handling of the messages we must create different elements that fit in the Pixelware environment, besides adding elements to the standard data model. In addition to the creation of the handling message mechanisms we must adapt the handled information to fit in our data model.

To carry out these tasks we create an independent module that will act as a link between Pixelware implementation and other European environments. To do this it is going to be used Spring Boot technology to create RESTful servers, the contratacion-core library, designed for the Pixelware data model representation in Java environment and some necessary third-party libraries for the message standardization such as ASIC-e container management and XML schema validation.

The way in which the implementation has been tested is by connectathon, an e-meeting of tests at European level with different centers which also worked on the project, separating the different tests in the agreed Use Cases. These consist of the accomplishment of a series of steps in which messages are sent to the other entities and the reception of the messages of the other entities is verified.

Keywords

electronic Tendering, Europe, XML, RESTful, standardization, UBL, SBD, validation, Pixelware, independent module, Spring Boot, ASIC-e, Schematron, connectathons

Agradecimientos

A todos mis compañeros en Pixelware, por plantearme y facilitarme la realización de este proyecto.

A mi profesor y ponente Francisco Saiz, por el interés que puso en este trabajo.

A mi familia y amigos, por el apoyo recibido durante toda la vida.

Al futuro, por todo lo que nos ha prometido.

INDICE DE CONTENIDOS

1 Introducción	15
1.1 Motivación	15
1.2 Objetivos	16
1.3 Organización de la memoria	17
2 Estado del arte	18
2.1 Licitación Electrónica de Pixelware	18
2.1.1 Pixelware Search – Tramitador Web	18
2.1.2 Portal de Licitación	18
2.2 Red de comunicación de e-Sens	19
2.2.1 SMP/SML	19
2.2.2 Gateway	19
2.2.3 Semántica de los mensajes	20
2.2.3.1 Contenedores ASiC y ASiC-e	20
2.3 Tecnología y lenguajes	21
2.3.1 Java + Maven + Git	21
2.3.2 RESTful, Spring y WINSW	21
3 Análisis	23
3.1 Requisitos del sistema	23
3.1.1 Requisitos Funcionales	23
3.1.1.1 Portal de Licitación	23
3.1.1.2 Pixelware Solutions	23
3.1.1.3 Pixelware Esens Message Receptor	24
3.1.2 Requisitos no funcionales	25
4 Diseño	26
4.1 Arquitectura	26
4.2 Modelo de datos	27
4.2.1 Mensajes e-Sens	27
4.2.2 Peticiones de Suscripción	27
4.2.3 Organismos de e-Sens	27
4.2.4 Modificaciones de datos	27
4.3 Estandarización del modelo y validación de documentos	28
4.3.1 Documentos UBL	28
4.3.2 Validación y documentos schematron	29
4.4 Comunicación por la red de e-Sens	30
4.4.1 Gateway	30
4.4.2 Enrutamiento	31
4.5 Empaquetamiento del mensaje	31
4.5.1 Standard Business Document	32
4.5.2 Contenedores ASIC-e y validación de certificados	32
5 Desarrollo	35
5.1 Librería central de contratación de Pixelware	35
5.2 Creación y manejo de documentos	36
5.3 Validación	36
5.3.1 Validación de contenedores	36
5.3.2 Validación de ficheros	37
6 Integración, pruebas y resultados	38
6.1 Connectathons	38

6.1.1 Contexto	38
6.1.2 Connectathon de octubre.....	39
6.1.2.1 Seguimiento	39
6.1.2.2 Resultado.....	39
6.1.3 Connectathon de noviembre	39
6.1.3.1 Seguimiento	39
6.1.3.2 Resultado.....	40
6.1.4 Connectathon de diciembre.....	40
6.1.4.1 Seguimiento	40
6.1.4.2 Resultado.....	40
6.1.5 Connectathon de enero.....	40
6.1.5.1 Seguimiento	40
6.1.5.2 Resultado.....	40
6.1.6 Connectathon de febrero	41
6.1.6.1 Seguimiento	41
6.1.6.2 Resultado.....	41
6.1.7 Connectathon de marzo.....	41
6.1.7.1 Seguimiento	41
6.1.7.2 Resultado.....	41
Conclusiones y trabajo futuro	42
6.2 Conclusiones	42
6.3 Trabajo futuro	42
Referencias.....	44
Glosario.....	47
Anexos	I
A Manual de instalación	I
B Diagrama de Casos de Uso	III
C Diagrama de Secuencia sobre un Expression of Interest	IV
D Anexo Pasos de Connectathon de octubre y su resultado.....	V
E Capturas de la Connectathon de marzo.....	VII

INDICE DE FIGURAS

FIGURA 1-1 BENEFICIOS DE LA LICITACIÓN ELECTRÓNICA	15
FIGURA 2-1 ESTRUCTURA SML/SMP	19
FIGURA 2-2 ESTRUCTURA DEL MENSAJE	20
FIGURA 2-3 ESQUEMA REST.....	21
FIGURA 4-1 ARQUITECTURA DEL PROYECTO E-SENS	26
FIGURA 4-2 REPRESENTACIÓN DE ENTIDADES EN UBL	28
FIGURA 4-3 ESQUEMA DE CONTENEDOR ASIC-E	33
FIGURA 4-4 TL DEL MINISTERIO DE INDUSTRIA, ENERGÍA Y TURISMO	34

FIGURA 5-1 ESQUEMA DAO..... 35

INDICE DE TABLAS

TABLA 2-1 LISTA DE GATEWAYS..... 20

TABLA 6-1 LISTA DE PARTICIPANTES DE LAS CONNECTATHONS..... 38

1 Introducción

En la siguiente sección viene redactada una introducción al contexto y los motivos del proyecto y su trasfondo, además de mostrar la organización del documento que se está leyendo.

1.1 Motivación

Gracias a las Tecnologías de Información y Comunicación, el sector de la Licitación Electrónica ha ido evolucionando hacia una representación cada vez más digital y menos analógica, facilitando el seguimiento de un contrato y las distintas acciones que se realizan sobre él. Aunque los actuales sistemas de licitación electrónica ofrecen ganancias de eficiencia sustanciales en comparación con la licitación tradicional basada en papel, en la mayoría de los casos todavía requieren que las empresas licitadoras tengan que aportar cierta información en diferentes formatos y sin permitir poder reutilizar la información obtenida.

Al existir un variable número de plataformas, los usuarios tienen que someterse a un proceso de aprendizaje para cada una de ellas, teniendo que añadir la información requerida en el formato correcto para cada una de ellas. Si bien este no es un problema significativo para los Organismos de Contratación, ya que suelen usar una misma plataforma, es una gran desventaja para los licitadores, que tienen que utilizar las plataformas elegidas por los distintos Organismos con los que pretenden hacer negocios.



Figura 1-1 Beneficios de la Licitación Electrónica

Una de las tareas más habituales y pesadas en este proceso es la preparación una respuesta a los distintos contratos que puede tener un expediente junto con las respuestas al pliego de condiciones. Si este proceso es complicado e incluso confuso en el ámbito nacional por la multitud de representaciones de los datos dependiendo de cada sistema de gestión de contratos, este mismo proceso se convierte en casi imposible cuando hablamos de ofertas a nivel europeo e internacional.

Para ello la solución presentada por la informatización de la licitación ofrece soluciones sencillas a problemas habituales a la hora de generar toda esa información, usando ciertos mecanismos para reducir el riesgo de error e incumplimiento de los parámetros requeridos de cada oferta. Los elementos que constituyen este tipo de ofertas se pueden simplificar y estandarizar para permitir un control de los datos eficiente de manera computacional. Pero no sólo los agentes de licitación son los beneficiados de esto sino también los distintos Organismos de Contratación pueden ver mejorados sus servicios reutilizando recursos

debido a la estandarización de las comunicaciones y ampliando el rango de sus ofertantes a empresas de los demás países de la UE.

Una solución para estos problemas es aplicar especificaciones estándar para las interfaces de las plataformas de Licitación Electrónica con el fin de poder intercambiar información de manera estándar para todas las plataformas de licitación electrónica y dar la oportunidad a empresas externas a realizar ofertas a contratos que hasta ahora se veían improbables. Con ello los proveedores de estas plataformas serían libres de diseñar sus soluciones, pero en todos los casos se convergería hacia un modelo estándar accesible para todos los licitadores.

1.2 Objetivos

Por el lado de los Organismos de Contratación se buscan los siguientes objetivos:

- Una mayor transparencia de los procedimientos de licitación para que los distintos licitadores puedan encontrar la oportunidad de negocio de manera sencilla.
- Incrementar el número de postores a cada oferta, pudiendo incluso llegar a ofertar empresas de otros países.
- Estandarizando sus datos, el proceso administrativo para evaluar las ofertas recibidas se realizaría de manera más sencilla e intuitiva.
- la garantía de la integridad de la oferta, ya que el éxito de estas depende directamente de los datos proporcionados por cada licitador y que la hará más o menos competitiva.

Por el lado de los licitadores se quiere cumplimentar los siguientes objetivos:

- Tener la garantía de la correcta validez de la oferta realizada y su correcta recepción en el plazo establecido.
- Asegurar un acceso equitativo a todas las ofertas publicadas, incluso en la participación de licitación entre varios países, en las que los criterios de clasificación y las evidencias son distintas. Para ello se desea usar aplicaciones y estándares compatibles con cualquier cadena de procesos de licitación.
- La reutilización de tanta información como sea posible, ahorrando tiempo y costes.
- El uso de información y documentos de licitación estandarizados también son factores a tener en cuenta para poder responder de manera rápida y con evidencias estandarizadas.
- Por último, otro factor importante es la recepción de documentos estandarizados para poder responder con evidencias estandarizadas.

En tercer lugar, los responsables políticos de los Estados miembros de la Unión Europea desean cumplir los siguientes objetivos:

- la reducción de la carga administrativa en la contratación pública.
- Interoperabilidad entre todas las soluciones de licitación para facilitar un solo mercado europeo, porque cuanto mayor sea la interoperabilidad, se amplía el grado de oferta.
- La posibilidad de la total compatibilidad de los procesos de licitación para distintas plataformas de licitación electrónica.

En el caso de la empresa Pixelware, las principales motivaciones que llevan a realizar este proyecto son:

- Ofrecer a sus usuarios la posibilidad de ampliar los horizontes a los posibles ejercicios de licitación en caso de los licitadores y a la amplitud de licitadores de la Unión Europea en el caso de los Organismos de Contratación.
- Realizar un proyecto de esta magnitud, conociendo los métodos de otros países y favoreciendo la cooperación entre distintas entidades.

1.3 Organización de la memoria

Esta memoria se estructura en secciones que describen todas las fases necesarias para entender el proceso de creación del proyecto mencionado. Cada sección está a su vez compuesta de distintas subsecciones que especificarán los conceptos más importantes de su respectiva sección. Este documento está compuesto por las siguientes secciones:

- **Estado del arte:** Se aportará la información relacionada con el estado actual de las implementaciones llevadas a cabo por Pixelware para el ámbito de la licitación y contratación electrónica y las principales características de las mismas. Además, se aportará datos del entramado de la red de e-Sens y de los distintos prerequisites que deberemos cumplimentar para poder llevar a cabo con éxito la funcionalidad del proyecto y la interconectividad con los demás participantes.
- **Análisis:** Se realizará un análisis de las distintas funcionalidades que deberá cumplimentar nuestra implementación de acuerdo con los distintos casos de uso planteados por e-Sens, todo ello teniendo en cuenta la correcta adaptación de estas funcionalidades al entorno de licitación electrónica de Pixelware, planteando además las distintas modificaciones de este último para amoldarse a los estándares necesarios.
- **Diseño:** En esta sección se explica el diseño y la estructura de los módulos que se han llevado a cabo para la creación de nuestro proyecto, además de comentar como está organizada toda la aplicación y cuales han sido los bloques a desarrollar. Por último, se indicarán las modificaciones a realizar en las distintas interfaces del entorno de licitación electrónica con el uso de maquetas.
- **Desarrollo:** Este apartado será dedicado al proceso de creación del código fuente del módulo, así como a las distintas modificaciones que realizaremos en la librería de contratación de Pixelware y las razones que nos motivan a realizarlas. También se aportará información sobre las distintas herramientas utilizadas para la correcta implementación del diseño planteado y sus principales características.
- **Integración, pruebas y resultados:** Se estudia el funcionamiento de la implementación del proyecto y las distintas opciones que se han realizado para probar su correcta operación. Además, mostramos los resultados de estas pruebas y la integración del módulo con los demás elementos del entorno de licitación electrónica de Pixelware.
- **Conclusiones y trabajo futuro:** Por último, se plantearán las conclusiones obtenidas de la implementación del módulo de comunicación con e-Sens y se introducirán las posibles ampliaciones o mejoras que se plantean en el futuro.

2 Estado del arte

En el siguiente apartado se planteará el estado actual del entorno de Licitación Electrónica de Pixelware y la consiguiente adaptación al modelo estándar europeo de eTendering planteado por e-Sens.

2.1 Licitación Electrónica de Pixelware

La empresa Pixelware lleva desde 1995 operando con expedientes de contratación, pero en los últimos años ha ido abarcando más fases del seguimiento de un contrato por lo que ha debido de adaptar sus distintas implementaciones y acoplarlas a las distintas fases de un contrato. Para ello se hace uso de dos de sus principales herramientas: El entorno de gestión de datos Back-End de Pixelware Solutions junto con el Tramitador Web, responsable de la representación del ciclo de vida de un expediente de contratación y sus distintos elementos y por otro lado el sistema Front-End llamado Portal de Licitación, creado para proporcionar a los distintos licitadores el acceso a la información de los expedientes y la interacción con ellos.

2.1.1 Pixelware Search – Tramitador Web

Pixelware Search es una aplicación Web encargada de mostrar y añadir registros asociados a fichas de una base de datos, la cual representa el modelo de datos del cliente en cuestión. En general, la aplicación aporta la funcionalidad de representación de documentos en formato electrónico, haciendo de interfaz para el acceso, creación, modificación y eliminación de registro de la base de datos. Además de las funciones esenciales de un motor de base de datos, Pixelware Search también permite indexar documentación a los distintos registros, relacionarlos entre si y realizar búsquedas a varios niveles, tanto locales por ficha como de manera general en la base de datos. Esta aplicación será la encargada de mostrar toda la información generada en el sistema.

Esta aplicación, junto con la aplicación de flujos de trabajo, se considera la base para el seguimiento de un expediente. El problema principal es que ambas aplicaciones llevan siendo usadas desde un gran periodo de tiempo, por lo que las implementaciones de ciertas funcionalidades necesarias para el seguimiento de un expediente se antojan complicadas. Para ello se usa el Tramitador Web, un módulo que aporta la funcionalidad a la que nos referimos, como puede ser la creación de distintos lotes para un mismo expediente o la apertura de sobres de un expediente. Junto con una librería de personalización de la interfaz HTML basada en JavaScript se logra realizar un seguimiento del expediente correcto e incluso realizar ciertas funciones del ciclo de vida de manera electrónica, como la presentación de ofertas, la apertura de sobres o el cálculo automático de la valoración de una oferta.

2.1.2 Portal de Licitación

El portal de Licitación de Pixelware es una aplicación Web basada en Java encargada de la interacción entre los licitadores y los distintos expedientes del Organismo de Contratación. En esta aplicación los licitadores podrán acceder a la información de los expedientes que alberga el Organismo e interaccionar con ellos, como realizando preguntas acerca del expediente, expresando su interés o por último procediendo a la realización de una oferta. Esta aplicación a su vez requiere autenticación por parte de los licitadores para realizar ciertas acciones sobre los expedientes llevada a cabo mediante certificado digital o usuario y contraseña.

2.2 Red de comunicación de e-Sens

En el siguiente apartado se verán descritos algunos de los puntos más importantes para conocer el funcionamiento de la red de mensajes de e-Sens y los distintos pasos a llevar a cabo para poder compartir mensajes entre las distintas entidades de manera correcta.

2.2.1 SMP/SML

La red conformada por e-Sens estaba basado en la conexión mediante Service Metadata Publishers y sus correspondientes servidores DNS llamados Service Metadata Locators. Al realizar una petición al servidor SMP, este realiza la búsqueda de información de manera local si la información está registrada en él o realiza una consulta de la localización de la información a los SML. Estos servidores están basados en la infraestructura usada por DNS, pero usando distintos tipos de registro y comunicándose mediante el protocolo ebMS3/AS4 basados en SOAP.

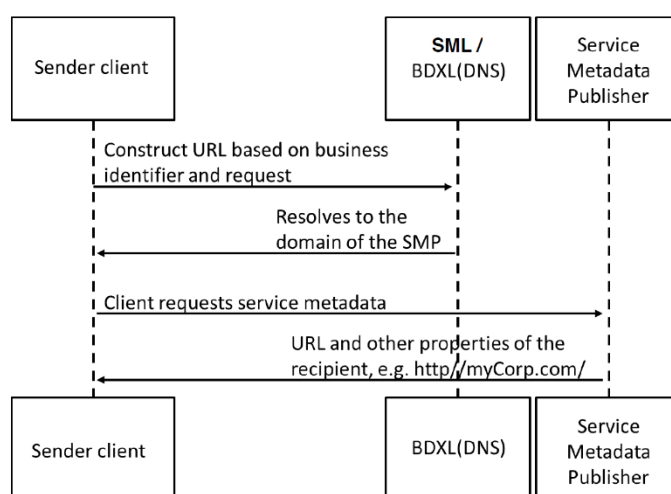


Figura 2-1 Estructura SML/SMP

2.2.2 Gateway

El Gateway es el elemento conector entre las distintas entidades que forman parte de la red de e-Sens. Es el elemento que se encarga de recibir los mensajes de los participantes que tiene registrados y de buscar a los demás gateways que tienen registrados a estos participantes para después enviarles el correspondiente mensaje. Este elemento abstrae la búsqueda de meta información de los distintos participantes y da un acceso simplificado para la compartición de información.

En nuestro caso vamos a utilizar el Gateway UPRC, localizado en Grecia. Este servidor usa las peticiones de HTTP para la interacción entre la información de los sistemas de licitación como entre los distintos participantes que constan en ese sistema de licitación. Toda la funcionalidad que llevaremos a cabo con el Gateway se realiza usando autenticación básica, por lo que para poder comenzar a compartir información primero debemos obtener un id de usuario y una contraseña para poder interaccionar con el Gateway.

Tabla 2-1 Lista de Gateways

Access Point	Tendering system(s) connected to the access point	Country
Holodeck B2B	TenderNed (2nd phase of the pilot)	Netherlands
IBM eC ServiceCenter	ETHICS	Denmark
	TenderNed (1st phase of the pilot)	Netherlands
UPRC AP	German Adapter (connects eVergabe)	Germany
	Pixelware	Spain
Oxalis	A majority of the access points governed by the Norwegian PEPPOL authority are using Oxalis	Norway

2.2.3 Semántica de los mensajes

Para el correcto envío de los mensajes, estos deberán llevar un formato determinado para que los servidores puedan procesar la información de manera correcta. Para ello usaremos el modelo de datos del Standard Business Document (SBD), los contenedores ASiC-E y los documentos UBL de representación de datos empresariales para formar el modelo standard de mensajes llamado e-Document.

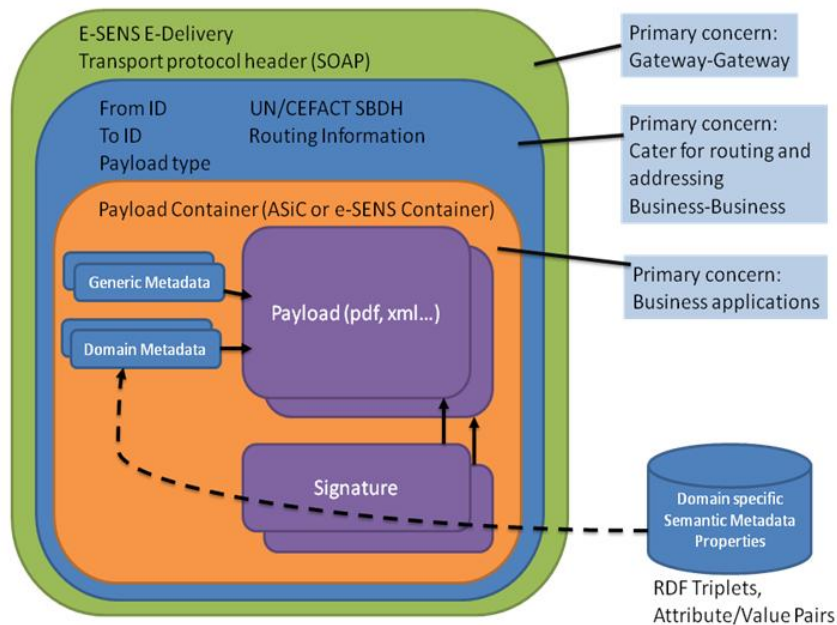


Figura 2-2 Estructura del mensaje

2.2.3.1 Contenedores ASiC y ASiC-e

En el cuerpo del SBD se encuentra el contenido del documento se envía, además de poder albergar otros documentos adicionales. Esta información deberá de ser contenida en un fichero firmado por la entidad que la ha creado para la comprobación de la identidad del emisor por parte de las demás entidades participantes. Para ello nos hacemos uso de la

tecnología de los ficheros ASiC, contenedores en formato ZIP que son asociados a una o varias firmas digitales y/o aserciones de tiempo.

2.3 Tecnología y lenguajes

A continuación, aportaremos información acerca de la tecnología que usaremos en la creación de este proyecto y las razones que nos han llevado a usar estos elementos en nuestro desarrollo.

2.3.1 Java + Maven + Git

De manera progresiva el uso de lenguaje Java se ha ido extendiendo en los desarrollos de Pixelware, sobre todo desde que la empresa ha ido representando más estados del ciclo de vida de un expediente de contratación. Java es un lenguaje de código abierto y basado en el paradigma de la programación orientada a objetos (POO) que pretende realizar representaciones virtuales a elementos reales y sus distintas interacciones. En nuestro caso tenemos claro los elementos que conforman el sistema por lo que la POO nos facilita la implementación y además los distintos módulos a modificar también se encuentran programados en lenguaje Java.

Además, se usará la herramienta Maven para la creación y gestión de proyectos en Java, ya que las distintas librerías a las que deberemos acceder están soportadas por Maven y además albergadas en un repositorio de la empresa. También mencionar que, por seguridad y control de versiones, se ha decidido usar Git como herramienta del control del versionado de nuestro software, usando también un servidor de repositorios Git de la empresa.

2.3.2 RESTful, Spring y WINSW

REST es un estilo de arquitectura basado en peticiones HTTP para obtener datos o indicar ejecuciones de operaciones sobre estos datos de manera similar a las instrucciones básicas de los sistemas de gestión de datos relacionales. Mediante esta arquitectura nos comunicaremos entre los distintos servidores, habiendo sido especificadas el tipo de peticiones a realizar en cada caso y las URI a las que se referenciaran. El hecho de compartir datos en formato XML beneficia el uso de REST para poder jerarquizar los mensajes, formando las distintas capas del protocolo.

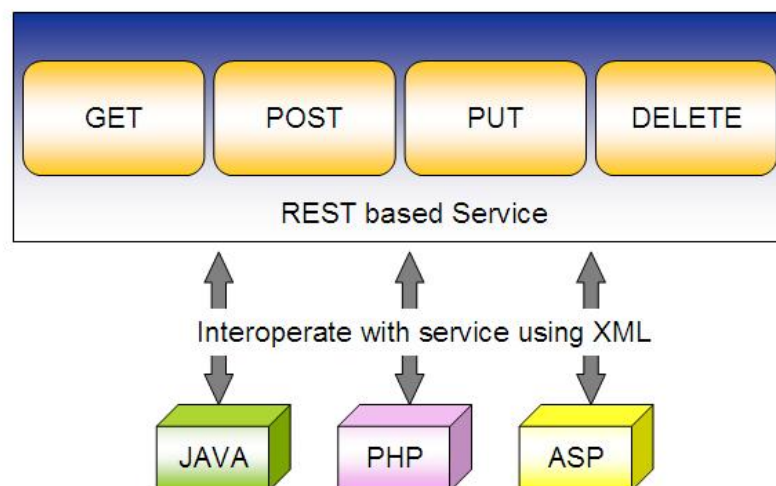


Figura 2-3 esquema REST

Para la creación de un servidor que use esta arquitectura se hará uso del Framework Spring. En particular de su Framework Spring MVC, una serie de librerías para facilitar la representación de datos mediante REST, además de dar soporte al envío de peticiones HTTP. Además de Spring MVC nos haremos uso de la funcionalidad Spring Boot, que genera de manera dinámica un servidor Apache exclusivo para el desarrollo creado, además de configurar de manera automática los servicios de monitorización y el contexto del framework.

En la mayoría de los casos los entornos de Pixelware funcionan en servidores con sistemas operativos de Windows, por lo que la creación del desarrollo como un servicio de Windows es una opción necesaria para controlarlo de manera más sencilla y rápida. Para ello usaremos la herramienta winsw, una aplicación basada en C# para el encapsulamiento y la creación de servicios de Windows para implementaciones en Java. Esta aplicación recibe una lista específica de parámetros y ejecuta una máquina Java que ejecute el desarrollo. Además de ello interconecta al SO con esta máquina Java, registrándola como un Servicio de Windows y pudiendo ser iniciada, parada y reiniciada en todo momento.

3 Análisis

En la siguiente sección aclararemos los distintos requisitos tanto funcionales como no funcionales marcados como necesarios para la correcta implementación de este proyecto. En primer lugar, enumeraremos los requisitos funcionales dependiendo de la aplicación que tendrá que cumplirlos y más tarde aportaremos los distintos requisitos no funcionales, esta vez agrupados en una sola sección.

3.1 Requisitos del sistema

A continuación, se planteará los distintos requisitos, especificando a qué módulo del sistema se aplica y cómo debe de comportarse el sistema para que se cumpla de manera adecuada.

3.1.1 Requisitos Funcionales

3.1.1.1 Portal de Licitación

El Portal de Licitación da acceso a los licitadores a la información de los expedientes y a la interacción con ellos, por lo que deberá ser modificado para permitir que los usuarios puedan interaccionar con expedientes externos. En este caso los licitadores deberán poder suscribirse a un expediente de contratación y solicitar información acerca del expediente. Si en algún momento las interacciones no hayan podido realizarse de manera correcta, se les será notificado por algún mensaje de alerta.

- I. **[PORTAL_LICITACION_RF1] Enviar una expresión de interés a un expediente externo:** Previa autenticación, el licitador registrado en el Portal de Licitación podrá suscribirse a un expediente de contratación de otro participante de e-Sens.
- II. **[PORTAL_LICITACION_RF2] Enviar una solicitud del estado de un expediente:** Previa autenticación, el licitador registrado podrá consultar el estado del expediente de contratación, si ha recibido respuesta o si, de lo contrario, ha habido algún error.
- III. **[PORTAL_LICITACION_RF3] Ver la documentación del estado de un expediente:** El usuario del Portal deberá poder acceder a ver la documentación obtenida mediante la solicitud del estado, siempre que sea posible su visualización.
- IV. **[PORTAL_LICITACION_RF4] Consulta del estado de las peticiones de suscripción realizadas:** El usuario del Portal deberá poder consultar el estado de las peticiones de suscripción realizadas, además de la fecha de envío.

3.1.1.2 Pixelware Solutions

La aplicación Pixelware Solutions da acceso a los miembros del Órgano de Contratación a la información correspondiente con el expediente y todas sus interacciones con los licitadores. En esta se podrá acceder a toda la información tanto de los expedientes como de los licitadores y los órganos de contratación cuyos expedientes se encuentren registrados en el sistema de e-Sens. Por ello la aplicación deberá mostrar la información almacenada acerca de las interacciones de los licitadores con los expedientes externos y las consiguientes interacciones entre los Órganos de Contratación de expedientes externos con estos y con los licitadores suscritos a estos. Además, la aplicación deberá albergar cierta información

adicional sobre estos expedientes y los licitadores que se pueden suscribir a estos para respetar el estándar impuesto por Esens.

- I. **[PIXELWARE_SOLUTIONS_RF3] Crear organismos externos a la plataforma:** el usuario de Pixelware Solutions podrá crear organismos externos asociándolos a una ID de e-Sens, pudiendo crear expedientes externos a la plataforma.
- II. **[PIXELWARE_SOLUTIONS_RF1] Crear expedientes externos a la plataforma:** el usuario de Pixelware Solutions podrá crear expedientes vacíos externos a la plataforma, que serán completados usando peticiones de documentación o siendo actualizado con documentación enviada por su organismo.
- III. **[PIXELWARE_SOLUTIONS_RF2] Crear licitadores externos a la plataforma:** el usuario de Pixelware Solutions podrá crear licitadores y anexarlos a una ID de e-Sens, pudiendo enviar mensajes por la red de e-Sens.
- IV. **[PIXELWARE_SOLUTIONS_RF4] Mostrar la lista de mensajes:** el usuario de Pixelware Solutions podrá ver los distintos mensajes, tanto enviados como recibidos, junto con información relevante, como su estado, el tipo de documento que transporta o, en caso de error, el mensaje de error correspondiente. Además, se deberá poder consultar tanto el mensaje enviado como los ficheros asociados.
- V. **[PIXELWARE_SOLUTIONS_RF5] Mostrar la lista de peticiones de solicitud:** el usuario de Pixelware Solutions deberá poder consultar las distintas peticiones de la solicitud de suscripción realizadas en el Portal además de ver el estado de estas.
- VI. **[PIXELWARE_SOLUTIONS_RF6] Mostrar la lista de licitadores y su identificación en la red de e-Sens:** el usuario de Pixelware Solutions deberá poder ver los licitadores que se encuentran registrados en la red de e-Sens.
- VII. **[PIXELWARE_SOLUTIONS_RF7] Mostrar la información de un expediente externo:** el usuario de Pixelware Solutions podrá consultar la información de un expediente externo que haya sido enviada por su Órgano de Contratación.

3.1.1.3 Pixelware Esens Message Receptor

Deberemos crear un servidor capaz de enviar mensajes a las distintas entidades que forman parte del proyecto de eTendering de e-Sens, recibir los mensajes que el Gateway alberga e interpretar estos mensajes para traducirlos al modelo de datos de Pixelware. En esencia, este módulo realizará un marshalling y unmarshalling de los elementos enviados en un caso y recibidos en otro. Además, deberá poder recibir peticiones de envío de mensajes a agentes externos mediante peticiones HTTP y albergar toda interacción de la información en la base de datos. En caso de fallo de cualquier índole, el sistema deberá guardar las causas de este.

- I. **[PIXELWARE_ESENS_RF1] Recepción de suscripciones de licitadores externos a un expediente:** El servidor deberá poder recibir peticiones de suscripción para un expediente, externo o no, y enviarlo si es posible o notificar al usuario si no ha sido posible.
- II. **[PIXELWARE_ESENS_RF2] Recepción de confirmación de suscripciones a un expediente de un organismo externo:** El servidor deberá poder recibir las confirmaciones de suscripción para un expediente y guardar la consiguiente información.
- III. **[PIXELWARE_ESENS_RF3] Recepción de peticiones de envío de solicitudes del estado de un expediente:** El servidor deberá poder recibir peticiones para el envío de solicitudes de estado para un expediente, externo o no, y enviarlo si es posible o notificar al usuario si no ha sido posible.

- IV. **[PIXELWARE_ESENS_RF4] Envío de suscripción de un licitador a un expediente de un organismo externo:** El servidor tendrá que realizar el envío de mensajes con las suscripciones de los licitadores a expedientes externos a la plataforma que hayan sido requeridas o, si no pueden ser enviadas, albergar la información de las causas del no envío.
- V. **[PIXELWARE_ESENS_RF5] Envío de confirmación de suscripción a un licitador externo:** Como respuesta a la solicitud de suscripción, el servidor deberá enviar un mensaje confirmando la suscripción al expediente, si ha sido posible.
- VI. **[PIXELWARE_ESENS_RF6] Envío de solicitud de documentación de expediente externo:** El servidor tendrá que realizar el envío de las solicitudes de documentación de un expediente que hayan sido requeridas o, si no han sido enviadas, albergar la información de las causas del no envío.
- VII. **[PIXELWARE_ESENS_RF7] Envío de documentación de un expediente:** El servidor deberá responder a la solicitud de documentación y enviar la correspondiente información sobre el expediente, además de los documentos publicables que estén anexados a este.
- VIII. **[PIXELWARE_ESENS_RF8] Recepción de mensajes:** El servidor podrá recibir mensajes de agentes externos enviados por el Gateway.
- IX. **[PIXELWARE_ESENS_RF9] Envío de notificación de error:** Si el servidor no consigue validar de manera correcta el mensaje, deberá enviar una respuesta con los distintos fallos surgidos en este proceso.

3.1.2 Requisitos no funcionales

- I. **[PIXELWARE_ESENS_RNF1] Seguridad y Privacidad:** El contenido de los mensajes sólo serán accesibles por los usuarios del sistema salvo que se establezca lo contrario. Además, las peticiones que se realizarán desde y hacia el servidor deberán ser autenticadas en todo momento, al igual que el propio contenido. En ningún momento se aceptará el contenido de un mensaje no firmado como válido.
- II. **[PIXELWARE_ESENS_RNF2] Extensibilidad:** El proyecto no está cerrado, ya que existen otro tipo de mensajes que pueden ser compartidos. Deberá crearse la aplicación de tal manera que el envío y la recepción de nuevos mensajes no modifique demasiado la implementación actual.
- III. **[PIXELWARE_ESENS_RNF3] Interoperabilidad:** Los mensajes que crearemos deberán ser interpretados por distintos sistemas con sus respectivos modelos y lenguajes. Los mensajes deberán de ser estandarizados para que todos los sistemas puedan operarlo de manera correcta.
- IV. **[PIXELWARE_ESENS_RNF4] Usabilidad:** el mecanismo para enviar mensajes deberá integrarse en las distintas aplicaciones Front-End de Pixelware y ofrecer de manera sencilla la posibilidad de enviar y recibir estos mensajes.

4 Diseño

En esta sección se realizará el análisis de los distintos requisitos planteados y cómo se acoplarán al entramado de licitación electrónica de Pixelware. Por separado se tratará de la arquitectura que constituyen los distintos módulos del proyecto y cómo interaccionan con las distintas implementaciones de Pixelware, el modelo de datos a seguir para respetar el entorno ya programado y las distintas modificaciones a realizar sobre él para la correcta robustez de los datos y por último los distintos cambios a realizar en las interfaces de las aplicaciones de Pixelware para cumplir los requisitos listados anteriormente.

4.1 Arquitectura

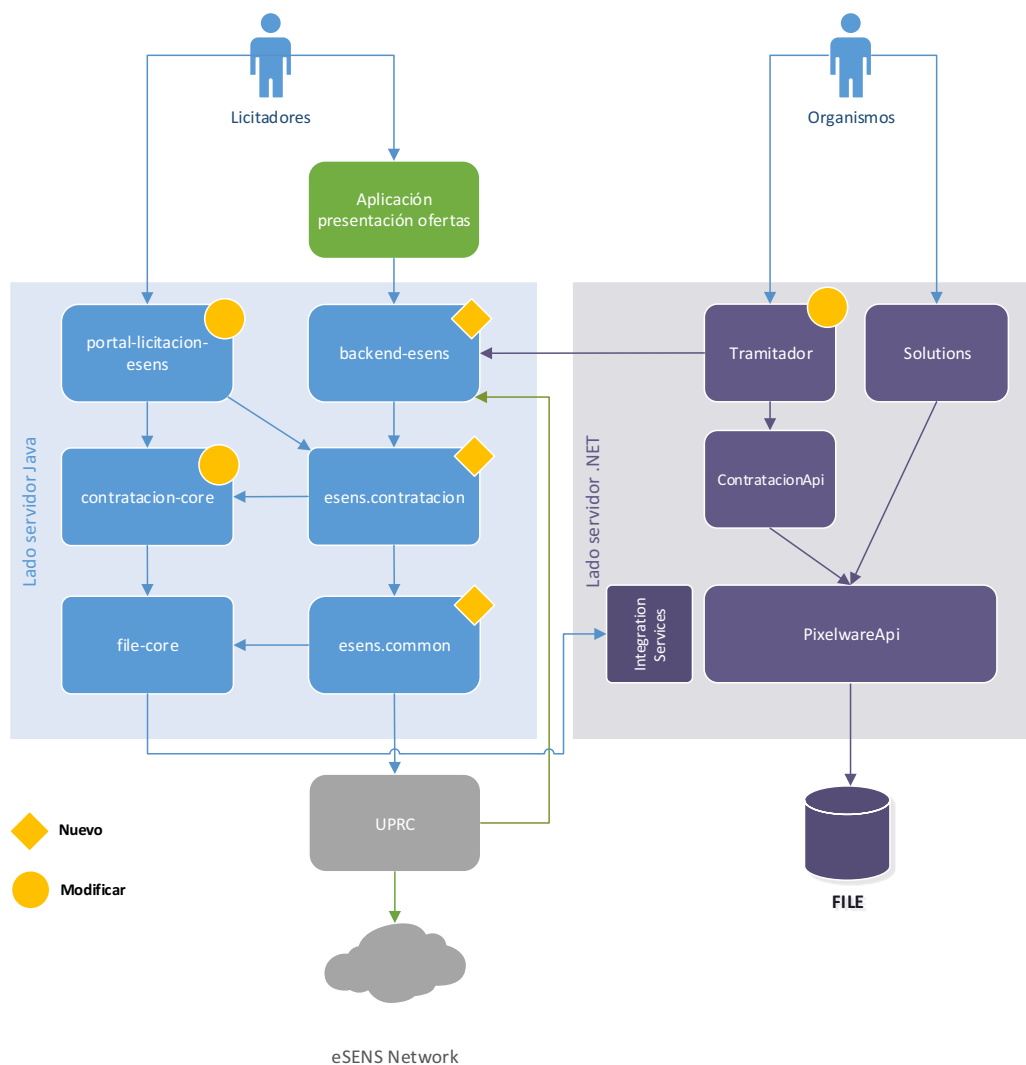


Figura 4-1 Arquitectura del proyecto e-Sens

Al tener un entorno de licitación electrónica ya implementado, es algo necesario que el módulo de interconexión de eSens se acople al entramado de aplicaciones que ya usa Pixelware. Para ello se plantea separar las distintas clases en tres módulos diferenciados que consten el núcleo del servidor de conexión con la red de eSens.

- esens.common: el módulo encargado del tratamiento de los documentos UBL, los contenedores ASiC-e y la validación de ambos. Además, se añadirán las clases auxiliares del servidor en este módulo, al ser el de más bajo nivel. En este módulo no se procesarán los documentos UBL, ya que en ningún caso este módulo manejará conceptos de licitación electrónica (aunque los documentos lo sean en esencia).
- esens.contratacion: Módulo central del servidor, encargado de la creación, envío y procesamiento de los mensajes.
- esens.backend: Módulo encargado del propio servicio de recepción de peticiones REST y de las distintas funcionalidades como servicio de Windows. En este caso nos basamos en las funcionalidades aportadas por el Framework de Spring y las clases necesarias para cumplir con los requisitos de winsw.

4.2 Modelo de datos

El modelo de datos que se tomará como referencia será el de Licitación Electrónica de Pixelware, en este caso la versión 3.1.1. Para adaptarlo a los requerimientos para la conexión del proyecto e-Sens, se añadirán algunos elementos necesarios al modelo.

4.2.1 Mensajes e-Sens

Para almacenar la información de los mensajes, tanto enviados como recibidos, deberemos añadir un elemento que los represente. Esta entidad albergará tanto la información de los participantes que intervienen en el mensaje, es decir, el emisor y el receptor de este, como la información del mensaje. Además, mantendremos un atributo para controlar el estado del mensaje, en caso de que no haya podido ser enviado correctamente, ya sea por problemas de procesamiento del mensaje, como errores en la conexión en la red o problemas con la validez del documento.

4.2.2 Peticiones de Suscripción

En el caso actual del modelo de datos 3.1.1. de Pixelware, no se tenía en cuenta que se pudieran hacer peticiones de suscripción, ya que la suscripción era un proceso automático, pero en el caso del proyecto de e-Sens pueden surgir problemas a la hora de validar o leer el documento, por lo que se deberá controlar el estado de una suscripción para controlar qué suscripciones han sido realizadas con éxito y cuáles no.

4.2.3 Organismos de e-Sens

El entorno de Pixelware ya permite que múltiples organismos publiquen las distintas ofertas de contrato en un mismo Portal, pero hasta ahora no se albergaba ninguna información acerca de estos. Para el proyecto de e-Sens deberemos albergar la información necesaria para poder enviar y recibir información acerca de los expedientes. Para ello debemos crear una entidad que represente los organismos y almacene esta información.

4.2.4 Modificaciones de datos

Para adaptar completamente nuestro modelo de datos deberemos realizar algunas modificaciones a las entidades para poder identificarlas en el entorno de e-Sens. Específicamente debemos agregar ciertos campos para almacenar los IDs usados en e-Sens para representar a los distintos elementos. En nuestro caso deberemos agregar información adicional al expediente y a la entidad de empresas, ya que ambas clases deberán albergar, por un lado, el ID de la publicación en el TeD, donde se publicarán los expedientes válidos

para el programa e-Sens, y por otro la identificación del entorno de e-Sens por parte de la empresa para que sea identificable y pueda enviar mensajes por la red.

4.3 Estandarización del modelo y validación de documentos

4.3.1 Documentos UBL

Para poder compartir la información de manera correcta debemos proponer un lenguaje estándar al que los participantes deban adaptarse. Para ello usaremos la librería UBL (Universal Business Document), creada por OASIS. Esta librería representa múltiples tipos de documentos de tipo empresarial en formato XML. En específico, la librería nos otorga la posibilidad de representar los distintos elementos y entidades de nuestro modelo de una manera sencilla y ya estandarizada.

```
<cac:TenderingTerms>
  <cbc:VariantConstraintIndicator>false</cbc:VariantConstraintIndicator>
  <cac:CallForTendersDocumentReference>
    <cbc:ID>bddd9596-7bfc-430e-b8c1-12fb32100dba</cbc:ID>
  </cac:CallForTendersDocumentReference>
  <cac:DocumentProviderParty>
    <cbc:EndpointID>http://www.tenderned.nl</cbc:EndpointID>
  </cac:DocumentProviderParty>
</cac:TenderingTerms>
<cac:TenderingProcess>
  <cbc:ProcedureCode listID="PR_PROC">1</cbc:ProcedureCode>
  <cac:TenderSubmissionDeadlinePeriod>
    <cbc:EndDate>2015-10-09+02:00</cbc:EndDate>
    <cbc:EndTime>14:13:45.579+02:00</cbc:EndTime>
  </cac:TenderSubmissionDeadlinePeriod>
  <cac:ParticipationRequestReceptionPeriod>
    <cbc:EndDate>2015-10-09+02:00</cbc:EndDate>
    <cbc:EndTime>14:13:45.579+02:00</cbc:EndTime>
  </cac:ParticipationRequestReceptionPeriod>
</cac:TenderingProcess>
<cac:ProcurementProject>
  <cbc:Name>Title of the works project</cbc:Name>
  <cbc:Description>Description of the works project</cbc:Description>
  <cbc:ProcurementTypeCode listID="PROJECT_TYPE">1</cbc:ProcurementTypeCode>
  <cac:MainCommodityClassification>
    <cbc:ItemClassificationCode listID="CPV">03113100-7</cbc:ItemClassificationCode>
  </cac:MainCommodityClassification>
  <cac:AdditionalCommodityClassification>
    <cbc:ItemClassificationCode listID="CPV">BA04-1</cbc:ItemClassificationCode>
  </cac:AdditionalCommodityClassification>
  <cac:RealizedLocation>
    <cbc:ID schemeID="NUTS">NL01</cbc:ID>
  </cac:RealizedLocation>
</cac:ProcurementProject>
```

Figura 4-2 Representación de entidades en UBL

Para poder transformar la información almacenada en el Portal a las distintas entidades propuestas por OASIS deberemos poder transformar nuestros distintos elementos a representaciones XML usando las distintas entidades de la arquitectura UBL. Para ello haremos uso de los paquetes de esquemáticos distribuidos por OASIS, los cuales se encuentra la información de las distintas entidades a usar por parte de los participantes de e-Tendering, eso es, las que representan un Expediente de Contratación y sus partes, además de los dos sujetos que interaccionan con ellos.

En el proyecto actual nos basaremos en la manipulación de cinco documentos de la librería UBL y, en concreto, con los documentos de interacción con un expediente. previa la adjudicación de este. Estos documentos son los siguientes:

- Application Response (MLR)
- Expression of Interest Request (EoI)
- Expression of Interest Response (EoIC)
- Tender Status Inquiry (TSI)
- Call for Tenders (CfT)

El mensaje Application Response es una respuesta en caso de error de validación. Este documento nos ayudará a realizar envíos sobre diagnósticos de la validación de los documentos que llegarán. Es un documento de soporte al uso de los distintos documentos UBL.

El mensaje de EoI nos servirá para realizar la petición de suscripción a un expediente externo. En este mensaje se albergará tanto la información del expediente al que se quiere suscribir como la propia información de licitador, para poder albergar la información de este. El mensaje de EoIC por otro lado servirá para confirmar la suscripción al expediente.

En nuestro segundo caso de uso usaremos el mensaje TSI para realizar la petición de documentación de un expediente externo al sistema. Este mensaje es necesario para que el sistema del Organismo nos envíe una última actualización de los detalles del expediente, además de la documentación. Esta información irá en el documento CfT, que además albergará los distintos documentos asociados al expediente y la clave necesaria para el cifrado de las ofertas que se realizarían al expediente.

4.3.2 Validación y documentos schematron

Además del sistema de envío y recepción de mensajes, el proyecto piloto de eTendering ha decidido realizar una previa validación de los documentos UBL que se reciben, para evitar errores en el posterior tratamiento del mensaje. La falta de datos obligatorios o la errónea cardinalidad de los elementos pueden dar problemas a la hora de transformar el documento en una representación correcta del expediente en nuestro modelo de datos.

Para realizar la validación de los documentos se usará la tecnología de los ficheros schematron. El Schematron es un lenguaje de validación XML que, mediante declaración de reglas y haciendo uso del lenguaje XPath puede comprobar la existencia o falta de patrones concretos en el árbol del fichero XML a validar. Usando uno o varios ficheros de validación, podremos resolver los problemas de la validación de los documentos UBL que pueden llegar a nuestro sistema. En estos ficheros encontraremos varias reglas para la validación del mensaje, clasificadas por bloqueantes, en el caso de que el fallo de validación sea grave y no se pueda realizar el procesamiento del mensaje, o no bloqueantes, en el caso contrario.

Además del registro de manera local, se plantea enviar un mensaje al emisor de un mensaje cuando este no logra pasar la validación de los ficheros schematron. Para ello haremos uso de los mensajes Message Level Response. En estos mensajes se encontrará detalladamente todas las reglas que el documento no ha logrado pasar y los motivos por los que se ha incumplido esa regla. Este tipo de mensajes será de envío obligatorio si el documento no pasa al menos una regla bloqueante, y recomendable si no pasa al menos una regla no bloqueante.

4.4 Comunicación por la red de e-Sens

Para poder compartir información con los demás participantes deberá existir un nexo entre ellos para poder enviarla por un medio seguro. Ahí es donde entra en escena la red de proyectos pilotos de e-Sens, entre los que se encuentran e-Tendering. A continuación, describiremos la arquitectura de la red y cómo es usada para transportar los distintos documentos.

4.4.1 Gateway

Para poder enviar los documentos por la red de pilotos deberemos primeramente encontrar el Gateway que mejor se adapte a nuestro proyecto. En el caso de la red de e-Sens, se encuentran distintos Gateways disponibles para la conexión. En el caso de Pixelware, por cercanía y por arquitectura, se ha determinado a usar el Gateway del centro de investigación de El Pireo, en Grecia.

Se ha escogido este Gateway por el método de compartir los mensajes que procesa, usando el modelo RESTful para hacer llegar los mensajes a los distintos participantes. Al contrario que los demás gateways, que hacían uso de representaciones SOAP de los mensajes para compartirlos, el modelo RESTful se adapta mejor a nuestro entorno, ya que facilita el envío y compatibiliza con las distintas librerías XML de las que se harán uso para construir los documentos a enviar.

Para poder conectarnos y poder enviar peticiones al servidor, primero se deberá registrar en el mismo, al necesitar autenticación para introducir mensajes en la red. Este procedimiento es necesario para que agentes externos no introduzcan información falsa. Para realizar el registro deberemos ponernos en contacto con los administradores del servidor de enrutamiento, en este caso, los miembros del equipo de investigación de la UPRC. El servidor mantiene una identificación de todos los servidores que se podrán conectar a él, cada uno con su correspondiente contraseña. A continuación, listaremos las distintas operaciones soportadas por el Gateway y con las que interactuaremos con él:

- Registro de participantes: el Gateway nos permite añadir nuevos participantes a la red e-Sens. Esto nos permite crear participantes asociados a un ID único que corresponderá con su ID en el envío y recepción de información. Para crear un participante se deberá crear un ID adecuado para la red de e-Sens, ya que no se acepta cualquier tipo de formato. Este ID será formado por dos elementos: El identificador del proyecto piloto y el identificador del propio participante. En el caso del piloto de e-Tendering, se usará el valor *iso6523-actorid-upis*. Este formato de identificación es el planteado por la ISO 15459 acerca la construcción de identificadores únicos.
- Registro de recepción de documentos: la red de e-Sens será usada no sólo por entidades de licitación sino por participantes de distintos ámbitos profesionales (Sanidad, Jurídicas, etc.). La recepción de documentos que no son de nuestro ámbito por algún inconveniente en el sistema no debe tenerse en cuenta como opción, ya que la información compartida en algunos casos puede ser confidencial. Para ello cada participante puede registrar los documentos que desea recibir, para que el propio sistema no permita el envío de tipos de mensajes que el receptor no acepta. Los documentos deberán ser representados por una identificación de su tipo de Documento y por una lista de proceso asociada al documento, ya que en algunos casos pueden usarse el mismo documento en ámbitos distintos.

- Borrado de registros: También se provee un método para el borrado de la información almacenada en el Gateway, tanto acerca los documentos a recibir como los propios participantes.
- Envío y recepción de mensajes: la principal función del Gateway deberá ser poder recibir y enviar mensajes a distintos participantes, ya estén usando el mismo servidor de enrutamiento para conectarse a la red o bien estén usando un Gateway distinto. Para ello tendremos un sistema de mensajes asíncrono en el que cada participante tendrá un buzón donde se almacenarán los mensajes que vaya recibiendo. Para la obtención de los mensajes del buzón de cada participante se plantean dos métodos:
 - Polling de mensajes: el Gateway se mantendrá a la espera hasta que el participante le solicite los mensajes. El propio participante deberá envía una petición con el ID del mensaje que quiere y este le responde con el mensaje en cuestión. Para ello el servidor también nos permitirá conocer el ID de todos los mensajes que se encuentran en el buzón.
 - URL de publicación: El gateway se encargará de enviar los mensajes que van llegando al cliente del participante. Para ello se deberá crear un receptor de peticiones REST en el cliente para permitir la recepción de mensajes por parte del Gateway. Este enlace se registra en el Gateway y va ligado al servidor del participante. Además, esta dirección deberá ser accesible sólo mediante autenticación para evitar la recepción de otro contenido que no provenga del servidor de enrutamiento.

4.4.2 Enrutamiento

Para conseguir un modelo de comunicación correcto deberemos identificar a los distintos participantes, de manera que pueda ser representado por un elemento de manera inequívoca. En el caso de la red de pilotos de e-Sens se plantean dos maneras de enrutamiento:

- Enrutamiento por entidad: cada licitador y Organismo de Contratación constituirían por si mismos un nodo en la red de Comunicación y los mensajes se intercambiarían de entidad a entidad. En los casos de países como Dinamarca o Alemania, donde sólo existe un único Portal de Contratación centralizado, este modelo se plantea beneficioso, ya que no se tiene pensado registrar a más de una entidad.
- Enrutamiento por participante: cada participante tiene un ID único para identificarse en la red de e-Sens. Los mensajes que envía este participante sólo podrán ser identificados por ese ID, ya sea de cualquiera de los licitadores u Organismo de Contratación que almacene la plataforma. Esta posibilidad se plantea debido a que en los países en los que existen diferentes Portales con distintos Organismos de Contratación, las entidades pueden cambiar de Portal de Licitación, o incluso tener expedientes en portales distintos.

Ya que tanto en España, por parte de Pixelware, como en Portugal, por parte de las empresas Vortal y Gatewit, existen distintos Portales, La consideración para el enrutamiento se debe plantear por participante por lo motivos mencionados anteriormente.

4.5 Empaquetamiento del mensaje

4.5.1 Standard Business Document

Para poder permitir el correcto redireccionamiento a realizar por los servidores Gateway deberemos añadir una capa a nuestro mensaje para identificar tanto quien lo envía como a quien se lo envía y qué le envía, al ser necesario un primer procesamiento por parte de los Gateway para identificar los distintos documentos que los participantes han registrado. En esta capa también deberá constar un identificador del tipo de documento y el contexto en el que se está usando, además de cierta información acerca del tratamiento del mensaje.

Todo esto se soluciona mediante el uso de un estándar de representación de datos llamado Standard Business Document o SBD. El SBD consta de una cabecera (SBD-Header) y de un cuerpo en el que se encuentra la información del documento. En el caso de e-Sens, la cabecera SBDH hará de cabecera de la capa de red, ya que en ella se albergará la información pertinente a los participantes para conseguir realizar el correcto redireccionamiento y que el mensaje llegue correctamente al receptor.

Como comentamos anteriormente, el fichero SBDH también puede transportar información útil para el participante, por lo que también deberá formar parte del cuerpo del mensaje. La información que alberga la cabecera engloba tanto la identificación del tipo de documento y su ámbito, así como si se debe responder con un mensaje de error en caso de fallo de validación del documento o si se debe de enviar un archivo REM al recibir el documento. Además de la información de los participantes, la cabecera albergará información de los distintos archivos que contiene y cierta información adicional tanto para el servidor Gateway como para el propio receptor del mensaje.

Por el otro lado, el consiguiente paquete de información que es transportado será albergado por el cuerpo del Standard Business Document, es decir, que el contenedor cifrado será el cuerpo de los mensajes que enviaremos.

4.5.2 Contenedores ASiC-e y validación de certificados

Como mencionamos anteriormente, debemos usar una estructura que nos permita albergar los distintos documentos que queremos enviar y que además nos permita asegurarnos del origen del mensaje. Para ello usamos los contenedores ASiC, que permiten agregar uno o varios certificados para la autenticación del emisor. La estructura de los contenedores ASiC incluye los siguientes elementos:

- Un directorio raíz que alberga la información correspondiente al documento y que puede incluir otros directorios que reflejen la estructura del contenido. En el directorio raíz se encontrarán los distintos elementos de información que se vayan a transportar, tanto el mismo documento UBL como los archivos adjuntos asociados a ese documento.
- Un directorio de información adicional al contenedor y metadatos. Este se encuentra albergado en el directorio raíz y que contiene metadatos del contenido, en el que se encuentran los tipos MIME de los documentos añadidos al contenedor, y la correspondiente firma o firmas que autentica o autentican el contenedor y a su creador.

En nuestro caso usaremos los contenedores ASiC Extended (ASiC-e) que permiten albergar información acerca de cada elemento del contenedor y realizar distintas firmas para varios conjuntos de información. Además de ellos usaremos el formato de firma estándar CAdES,

frente al formato XAdES, usando específicamente un certificado digital X.509 para la autenticación del emisor.

El objetivo de los certificados enviados en los mensajes es que el receptor pueda garantizar el origen del mensaje, por lo que se deberá comprobar la validez del certificado antes de procesar la información que se encuentra en el interior. Para ello deberemos diseñar un módulo del proyecto que se encargue de comprobar la validez de estos certificados. Pero para poder validar los certificados deberemos conocer el certificado o comprobar su validez comprobando la de las autoridades certificadoras que corroboran el certificado. Al ser un proyecto piloto, hemos decidido usar certificados auto-firmados y mantener un repositorio de los certificados para poder verificarlos. Para ello hemos usado dos métodos para poder mantener un almacén de certificados.

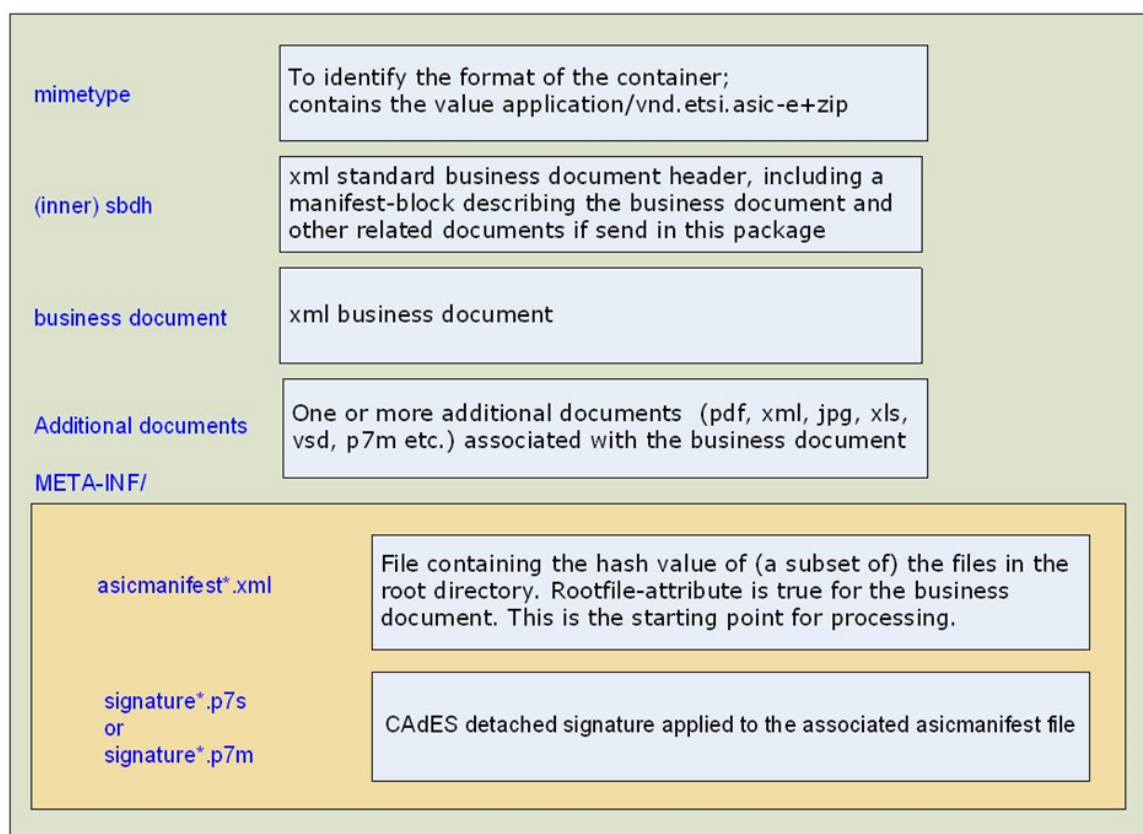


Figura 4-3 Esquema de contenedor ASIC-e

- Almacén de llaves: Para los primeros pasos del proyecto se decidió usar un almacén de llaves que albergase la información de todos los certificados de todos los participantes. Para validar la información usando el almacén de llaves sólo se deberá comprobar si el certificado recibido es primeramente válido y si se encuentra dentro del almacén. Este almacén es mantenido de manera local por cada participante, por lo que se deberá mantener actualizado en cada uno de ellos.
- Lista de Confianza: Como solución para no tener que mantener un almacén de llaves en cada participante, dando probabilidad a redundancia de datos y errores causados por ella, se decide usar una Lista de Confianza común para todos los participantes.

Esta lista es un fichero XML que contiene toda la información de un grupo de certificados validos usados para un fin específico. En nuestro caso, la lista de confianza albergará la información de los certificados que los participantes usarán durante la duración del proyecto. El documento se encontrará en un proveedor de servicios de confianza y, al igual que el almacén de llaves, se deberá comparar el certificado de la lista con el del mensaje original.

Además, en el caso de la Lista de Confianza, el documento que la alberga además información de la propia lista, como el certificado que autentica al creador de la lista o una fecha límite de la validez del documento. Usando esta fecha se podrá implementar un sistema de cacheo para poder mantener una versión actualizada del documento en memoria y no realizar llamadas constantes al servicio.

Se toma como comienzo la implementación de la validación mediante almacén de llaves, ya que no requiere realizar la conexión con el proveedor de validez, además de que en principio los participantes son pocos y la caducidad de los certificados es lo suficientemente grande para no ser un percance en el proyecto. Aun así, se ha ofrecido la oportunidad de la implementación por listas de Confianza, ya que deberá ser el método a usar al final del proyecto.

```
<OtherTSLPointer>
  <ServiceDigitalIdentities>
    <ServiceDigitalIdentity>
      <DigitalId>
        <X509Certificate>
          MIIIDyzCCAr0GAWIBAgIJAL6B32RxtXxMMA0GCSqGSIb3DQEBCwUAMH4xKzApBgNVBAMTI1NQQU5JU0ggVFJl
        </X509Certificate>
      </DigitalId>
    </ServiceDigitalIdentity>
    <ServiceDigitalIdentity>
      <DigitalId>
        <X509Certificate>
          MIIIDyzCCAr0GAWIBAgIJAKWF81P5Bqy1MA0GCSqGSIb3DQEBCwUAMH4xKzApBgNVBAMTI1NQQU5JU0ggVFJl
        </X509Certificate>
      </DigitalId>
    </ServiceDigitalIdentity>
  </ServiceDigitalIdentities>
  <TSLLocation>
    https://sede.minetur.gob.es/Prestadores/TSL/TSL.pdf
  </TSLLocation>
  <AdditionalInformation>
    <OtherInformation>
      <TSLType>
        http://uri.etsi.org/TrstSvc/TrustedList/TSLType/EUGeneric
      </TSLType>
    </OtherInformation>
    <OtherInformation>
      <SchemeTerritory>ES</SchemeTerritory>
    </OtherInformation>
    <OtherInformation>
      <ns3:MimeType>application/pdf</ns3:MimeType>
    </OtherInformation>
    <OtherInformation>
      <SchemeOperatorName>
        <Name xml:lang="en">MINISTRY OF INDUSTRY ENERGY AND TOURISM</Name>
        <Name xml:lang="es">MINISTERIO DE INDUSTRIA ENERGIA Y TURISMO</Name>
      </SchemeOperatorName>
    </OtherInformation>
    <OtherInformation>
      <SchemeTypeCommunityRules>
        <URI xml:lang="en">
          http://uri.etsi.org/TrstSvc/TrustedList/schemerules/EUcommon
        </URI>
        <URI xml:lang="en">
          http://uri.etsi.org/TrstSvc/TrustedList/schemerules/ES
        </URI>
      </SchemeTypeCommunityRules>
    </OtherInformation>
  </AdditionalInformation>
</OtherTSLPointer>
```

Figura 4-4 TL del Ministerio de Industria, Energía y Turismo

5 Desarrollo

A continuación, se ampliará la información acerca del proceso de Desarrollo del módulo Pixelware eSens Message Receptor. En ella se aportará toda la información en el apartado de la creación de los módulos anteriormente diseñados.

5.1 Librería central de contratación de Pixelware

En primer lugar, deberemos aplicar los cambios necesarios a los elementos que Pixelware ya ha implementado y que forman parte del modelo de datos de la empresa. En nuestro caso deberemos añadir las distintas clases mencionadas anteriormente para la adaptación de los datos, además de añadir ciertos atributos o métodos a clases ya existentes para completar la funcionalidad deseada. En particular, deberemos crear las clases que modificarán tanto a los expedientes de contratación como las distintas maneras de que un licitador se inscriba en el expediente para recibir actualizaciones.

Para ello y siguiendo con el modelo de desarrollo de Pixelware se crearán dos clases fundamentales que controlarán un objeto: Por un lugar la clase representativa del objeto y por otro la clase encargada de buscar, cargar y guardar las distintas entidades albergadas en la base de datos, o también llamada DAO (Data Access Object). Con esta metodología separamos la funcionalidad intrínseca del objeto de su interacción con los sistemas de almacenamiento.

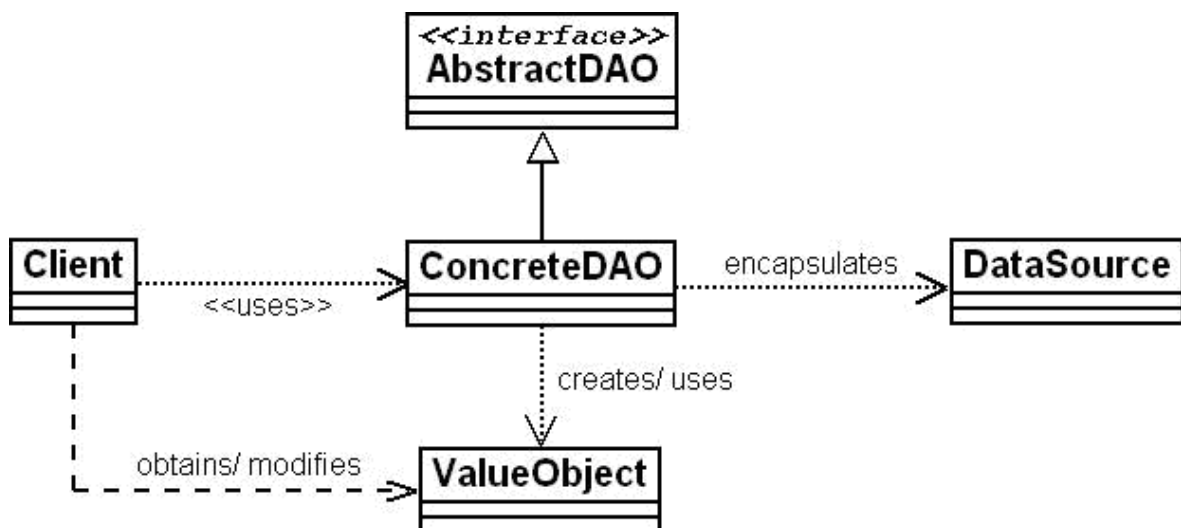


Figura 5-1 esquema DAO

En el caso particular de la representación de los mensajes se ha creado una clase MensajeEsens en la que la información del mensaje se verá representada (ID del mensaje, emisor, receptor, fecha y hora de llegada, documentos adicionales o representativos del mensaje) y por otro lado se creará una clase MensajeEsensDAO que será la encargada de cargar los mensajes que ya hayan sido creados anteriormente, buscarlos mediante su ID o

guardar nuevos mensajes que vayan entrando. Incluso podremos modificar los mensajes que se encuentren guardados por si ha habido algún fallo al procesarlo, tanto por parte del módulo de Pixelware como por parte de módulos de otros participantes.

Las principales modificaciones a realizar se basan en añadir atributos necesarios para poder compartir la información con el exterior, como pueden ser los ID de participantes de e-Sens en el caso de las empresas registradas. En estos casos, en vez de modificar la librería estándar, se ha decidido crear una clase DAO que complemente a la clase principal para obtener los valores específicos.

5.2 Creación y manejo de documentos

Para poder enviar correctamente los mensajes por la red de e-Sens deberemos crear una serie de representaciones de datos acorde a las representaciones UBL. Haciéndonos uso de la librería de entidades UBL generada a partir de sus esquemáticos, podremos representar las entidades de nuestro modelo, por lo que nuestro objetivo principal es crear las distintas clases que se encargaran de crear objetos de estas entidades y los transformaran en representaciones de nuestro modelo, además de poder realizar la transformación de manera inversa, como pueden ser los distintos stubs de un proceso de Marshalling-Unmarshalling.

Para este proceso, deberemos realizar dos tipos de transformaciones, en ambos sentidos: Deberemos poder transformar la representación de las entidades en formato XML que llega por los mensajes en representaciones UBL y además deberemos poder transformar estas entidades en las de nuestro modelo de datos para poder almacenarlas e interactuar con ellas. Por lo tanto, se han implementado dos clases que realizan estas transformaciones: La clase EsensFactory, que será la encargada de las transformaciones desde las entidades UBL a las entidades Pixelware y viceversa; y la clase UBLFactory, que será la encargada de transformar las representaciones UBL en ficheros XML con sus datos.

5.3 Validación

Para llevar a cabo la validación de los distintos elementos usaremos una clase llamada EsensValidator, que se encargará de realizar tanto las validaciones de los ficheros XML que llegan al servicio como los contenedores en los que se encuentra y su autenticación.

5.3.1 Validación de contenedores

Para la comprobación de los certificados que firman los contenedores ASiC-E se crearán varios métodos para los distintos casos que nos encontraremos. Antes de explicar las implementaciones realizadas, debemos explicar los motivos por los que se realizan. Al realizar un proyecto en paralelo con otros participantes europeos, las tecnologías y los métodos para la creación de los servicios es muy variada, por lo que puede haber algún escollo que se debe salvar para poder seguir con el desarrollo. En este caso nos encontramos con el problema de creación de contenedores, ya que el participante Vortal realiza su proyecto en lenguaje C# y, por tanto, no ha podido hacer uso de la librería de creación de ASiC-E propuesta por TenderNed e implementada en Java. El problema reside en que el algoritmo de compresión de datos que usa la implementación de este participante no es permitido por la librería en Java que se usa. Por ello, se ha decidido no usar la validación que ofrece la librería e implementar una propia usando la librería de ficheros ZIP4J. Con ella podemos extraer el certificado del contenedor y poder validarlo.

Como ya mencionamos anteriormente, las validaciones de los certificados se podrán realizar mediante comprobación de un almacén de llaves local o mediante consulta a una Lista de

Confianza, por lo que se deberán implementar ambas opciones. En primer lugar, se implementa la validación de los certificados mediante comprobación en almacén local, mediante la librería de gestión de certificados de Java. Por otro lado, para la validación por Lista de Confianza primero comprobaremos si tenemos una lista ya cacheada y si es válida y, en caso contrario, la actualizaremos y después obtendremos la representación del correspondiente certificado en Base64.

Por último, mencionar la creación de la clase EsensCert para la manipulación tanto del certificado con el que firmamos los contenedores como del almacén de llaves donde se encuentran las claves de los demás miembros.

5.3.2 Validación de ficheros

Por otro lado, la clase EsensValidator también realiza la validación de los documentos UBL, haciendo uso de la librería de manipulación de ficheros de reglas schematron ofrecido por el compañero Philip Helger. Para realizar esta validación se implementan dos métodos: un método que obtendrá el resultado de la validación y otro que obtendrá el detalle de la validación. Este planteamiento se realiza para no obligar a la validación a cargar un fichero de detalle de la validación cada vez que realizamos una validación. En cambio, sólo se obtendrá un detalle de la validación si esta ha sido fallida.

Por otro lado, dentro de las cabeceras de los paquetes SBD nos encontramos un atributo que mide el nivel de validación, pudiendo ser simple o estricta. Mediante la lectura del parámetro del SBDH, podemos comprobar cuán permisivo quiere que sea el participante con su documento.

6 Integración, pruebas y resultados

En la siguiente sección comentaremos las distintas pruebas realizadas para el correcto funcionamiento del módulo realizado.

6.1 Connectathons

Las principales pruebas realizadas sobre este módulo se basan en connectathons. Un connectathon es un evento centrado en una especificación de interoperabilidad construida por consenso abierto. El propósito del connectathon es tanto probar que la especificación es correcta como probar que las distintas implementaciones de esa especificación pueden conectarse y comunicarse entre ellas. El principio más importante de un connectathon es que es un lugar seguro para el fracaso en estos esfuerzos, es decir, que está libre de consecuencias negativas de un error en la implementación de alguien y que la especificación podría necesitar ser refinada. Para este proyecto se han realizado tres connectathons correspondiéndose con los meses de octubre, noviembre y diciembre del año 2016 y tres correspondiéndose con los meses de enero, febrero y marzo del año 2017.

6.1.1 Contexto

En las distintas connectathons deberemos realizar pruebas de los casos de uso implementados y demás funcionalidad contra las implementaciones creadas por los otros miembros. Los miembros que formarán parte de las connectathons en los casos de uso que nos corresponden son los siguientes:

- Pixelware, sistema de licitación electrónica de España en el ámbito tanto de licitador como de Órgano de Contratación.
- TenderNed, sistema de licitación electrónica del gobierno de Holanda, en el ámbito de licitador como de Órgano de Contratación.
- Vortal, plataforma de licitación electrónica de Portugal, en el ámbito tanto de licitador como de Órgano de Contratación.
- e-Vergabe, plataforma de licitación electrónica de Alemania en el ámbito de Órgano de Contratación.
- ETHICS, sistema de licitación electrónica de Dinamarca en el ámbito tanto de licitador como de Órgano de Contratación.

Para la realización de estas pruebas realizaremos una conexión en grupo usando la herramienta online Adobe Connect para la comunicación entre los distintos representantes de las entidades. Además, en cada connectathon se creará un planning en los que se deberán seguir las pruebas paso a paso y, en caso de que se tenga que detener o no realizar un paso, se debe argumentar el motivo.

Tabla 6-1 Lista de Participantes de las Connectathons

Participating MS	Etendering system (C1/C4)	Access point (C2/C3) and transport protocol
Netherlands	TenderNed	Holodeck B2B AS4
Denmark	ETHICS	IBM (AS2 and A42)

Germany	eVergabe	Adesso (Holodeck B2B AS4 / Oxalis AS2)
Portugal	Gatewit	eSPap (Holodeck B2B AS4 / Oxalis AS2)
Portugal	Vortal	Holodeck B2B AS4
Spain	Pixelware	UPRC (Flame AS4 / Oxalis AS2)

En el Anexo C y D pueden ver los resultados obtenidos en la última connectathon del proyecto, la correspondiente con el mes de marzo.

6.1.2 Connectathon de octubre

6.1.2.1 Seguimiento

Se realizan diferentes pasos en los que un miembro del connectathon debe mandar un mensaje y comprobar que el mensaje ha sido enviado correctamente. Por otro lado, el miembro receptor deberá confirmar o denegar su recepción y comprobar si el mensaje ha sido procesado correctamente o si por el contrario ha habido algún error.

En el caso de Pixelware se deben realizar siete pasos para realizar con éxito las pruebas de la reunión, siendo cuatro de ellas envío de expresiones de interés y las tres restantes respuestas de confirmación de expresión de interés previamente enviadas por los demás miembros.

6.1.2.2 Resultado

En el momento de la prueba se encontró que se había modificado la respuesta del servidor Gateway y además se realizaban envíos tanto de peticiones HTTP de múltiples partes (multipart/form-data) como peticiones de parte única. Estas distintas modificaciones no permitieron realizar el push de los mensajes que llegaban desde el Gateway a nuestro servidor por lo que se tuvo que realizar el polling de los mensajes de manera manual e introducirlo en el sistema mediante un cliente de peticiones HTTP.

Aun así, el resultado de las pruebas se antoja positivo, pudiendo realizar de manera correcta seis de los siete pasos a realizar. El paso no llevado a cabo se corresponde con la respuesta a la expresión de confirmación de ETHICS, al faltarles datos obligatorios sobre el licitador interesado

6.1.3 Connectathon de noviembre

6.1.3.1 Seguimiento

Al igual que en el anterior Connectathon realizaremos envíos de expresiones de interés y contestaremos a las que nos lleguen. Además, también nos disponemos a enviar respuesta de la aplicación en el caso de que los mensajes entrantes no pasen la validación de manera limpia. Esto, por otro lado, no comprueba que el mensaje no pueda ser procesado, sino que puede tener errores leves.

En este connectathon también podemos mostrar las distintas funciones agregadas al portal de licitación para permitir realizar expresiones de interés de manera online y pulsando tan solo un botón.

6.1.3.2 Resultado

Al igual que la connectathon anterior, los resultados de todos los pasos son positivos, pudiendo enviar correctamente las expresiones de interés y recibiendo las respuestas de manera correcta. Por otro lado, el envío de respuestas de la aplicación sólo pudo ser capturado por el participante eVergabe ya que los demás aún no habían registrado el tipo de mensaje en su participante de la red de e-Sens.

6.1.4 Connectathon de diciembre

6.1.4.1 Seguimiento

En esta Connectathon comenzaremos a realizar las pruebas del caso de uso de la solicitud del estado del expediente además de probar las implementaciones realizadas anteriormente. Además, este es el primer Connectathon en el que se realizarán pruebas contra el sistema de ETHICS, por lo que se deberá poner especial atención al resultado de los test que se realizan contra ellos.

6.1.4.2 Resultado

La Connectathon se puede catalogar como positiva, ya que la mayoría de pruebas en las que se ha participado han sido correctas. Nos encontramos con problemas realizando las pruebas contra ETHICS, tanto en la recepción de mensajes de respuesta creados por nuestro servidor, causado por la validación fallida del SBDH exterior, como a la hora del envío de los mismos mensajes, en este caso causado por un error de SMP entre los gateways. Estos problemas fueron insalvables durante la Connectathon por lo que no se pudieron realizar estos pasos.

Por otro lado, las pruebas de envío y recepción de los mensajes relacionados con el nuevo caso de uso fueron favorables. Al igual que en los anteriores pasos, también nos encontramos problemas entre los Gateways a la hora de recibir el CallForTenders del participante ETHICS. Además, su sistema no puede enviar un TSI sin que el licitador no se haya registrado al expediente, por lo que esos pasos no serán testeados.

6.1.5 Connectathon de enero

6.1.5.1 Seguimiento

En esta Connectathon volveremos a realizar pruebas de ambos casos de uso, además de probar la nueva funcionalidad de validación de los documentos UBL y envío y recepción de MLR en caso de error de validación de un documento.

6.1.5.2 Resultado

En los pasos necesarios de la Connectathon con respecto al caso de uso de expresión de interés, todos los pasos fueron realizados correctamente salvo por el sistema de ETHICS. Los pasos en los que estaba involucrado este participante han sido fallidos por el uso de una versión obsoleta de los ficheros de reglas para la validación y un problema relativo al Gateway. En primer lugar, el Expression of Interest enviado por ETHICS provocó un fallo de validación fatal con su correspondiente envío de MLR. En segundo lugar, el envío del Expression of Interest Confirmation desde ETHICS a Pixelware no pudo ser realizado por un fallo relacionado con la recepción del mensaje en el Gateway.

Con respecto a los pasos que conciernen al caso de uso del envío de la información del Expediente, los pasos con el sistema de ETHICS no pudieron ser cumplimentados, ya que la implementación del sistema no permite realizar envíos de TSI si no se ha podido hacer antes la suscripción al expediente. Por otro lado, el envío del CallForTender por parte de eVergabe no pudo completarse por incongruencias en los datos de enrutamiento.

6.1.6 Connectathon de febrero

6.1.6.1 Seguimiento

En esta Connectathon continuamos con la realización de pruebas de ambos casos de uso y el envío de MLR en caso de fallo de validación, tras haber implementado la modificación para el enrutamiento mediante Endpoints y no mediante entidades. En cualquier caso, se pospone el testeo del uso de ids distintas de Endpoint y entidad.

6.1.6.2 Resultado

En los pasos necesarios de la Connectathon con respecto al caso de uso de expresión de interés, todos los pasos fueron realizados correctamente salvo por el sistema de ETHICS. Los pasos en los que estaba involucrado el sistema de ETHICS han sido fallidos por un problema relativo al Gateway. En ambos casos, los envíos de los mensajes tanto de expresión de interés como de la confirmación de la expresión de interés no han podido ser enviados por esta causa.

Con respecto a los pasos que conciernen al caso de uso del envío de la información del Expediente, los pasos con el sistema de ETHICS no pudieron ser cumplimentados por el mismo motivo que en la anterior Connectathon. Por otro lado, el envío del CallForTender por parte de TenderNed a Pixelware no se pudo realizar debido a un error interno de TenderNed con el ID interno del expediente.

6.1.7 Connectathon de marzo

6.1.7.1 Seguimiento

Como última Connectathon oficial del Proyecto de e-Sens, su objetivo es probar todos los casos en los que participa nuestro proyecto, tanto los casos satisfactorios como los erróneos. Además, se probará la validación de certificados mediante Lista de Confianza, sustituyendo a la validación provisional mediante almacén de llaves.

En esta última Connectathon ETHICS decidió no participar, por lo que se deberá tener en cuenta que no se realizaron pruebas contra su implementación.

6.1.7.2 Resultado

En este Connectathon no se obtuvieron resultados negativos en ningún momento del envío y/o recepción de mensajes. Además, el envío de mensajes erróneos de manera deliberada también respondió de manera correcta, ya que todos los miembros enviaron su respuesta de error al correspondiente miembro.

Conclusiones y trabajo futuro

6.2 Conclusiones

Mediante la cooperación entre varias entidades separadas en distintos países y el apoyo de los organismos conformados en la Unión Europea, Pixelware ha podido realizar una integración exitosa a distintos centros de contratación para la compartición de información de expedientes.

Con la implementación realizada hasta ahora los usuarios de la plataforma de Licitación Electrónica que ofrece Pixelware, los licitadores pueden encontrar expedientes externos a la plataforma sin ningún tipo de trabajo extra. Además, podrá obtener información acerca de estos y suscribirse a él, en el caso de que el expediente sea modificado o cambie de estado en su ciclo de vida.

6.3 Trabajo futuro

Al final del proyecto los participantes han llegado al acuerdo de seguir realizando Connectathons a final de cada mes para probar algunas implementaciones o modificaciones de los servicios de los participantes. Esto nos permitirá seguir realizando pruebas sobre las distintas características que se pueden añadir al proyecto.

Por otro lado, el Gateway que usa el sistema de Pixelware para la red de e-Sens está implementado por el centro de investigación de la Universidad de El Pireo, Grecia. Al ser un proyecto de investigación a nivel universitario, los miembros del centro nos informaron que dejarán de dar servicio de Gateway al final del proyecto, pero nos permitirán usar sus recursos e información para usar su implementación e implementarlo nosotros mismos. En un futuro se deberá implementar un Gateway similar al implementado por UPRC e incluso poder dar servicio a otros participantes a conectarse a la red de e-Sens.

El siguiente paso lógico a dar sería la implementación de un último caso de uso: el envío de ofertas. Este caso cerraría el ciclo de interacción de un expediente con un licitador hasta la fecha de apertura de ofertas. Para ello se deberían poder recibir ofertas externas al sistema y agregarlas a las ofertas ya realizadas. Además, se deberá comprobar que las ofertas tienen una correcta validez. Del mismo modo, también se puede agregar la funcionalidad para que un licitador se pueda desuscribir de un expediente externo o permitir eliminar su ID de la red de e-Sens.

Otra posibilidad de cambio es la modificación de la herramienta Tramitador Web para dar la posibilidad de que los propios miembros del Organismo de Contratación puedan enviar el estado de un expediente sin tener que recibir una petición o de recibir notificaciones cuando se produzca una respuesta a la solicitud enviada por el licitador.

Referencias

- [1] eSens eTendering Pilot documentation, Jerry Dimitrou, Fokion Logothetidis. 2017. <http://wiki.ds.unipi.gr/display/ESENSPILOTS/D5.6-1+-+5.1.1+-+eTendering>
- [2] UPRC Gateway documentation, Jerry Dimitrou y Melis Ozgur Cetinkaya Demir, 2017, <http://wiki.ds.unipi.gr/display/ESENS/SBB++CIPA+Backend+Connector+with+AP+and+SMP+-+1.0.0>
- [3] Universe Business Document (UBL), OASIS, https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=ubl
- [4] ISO Schematron, Rick Jelliffe, 2006 <http://schematron.com/an-overview-of-schematron/>
- [5] Associated Signature Container (ASiC) Documentation, VEFA, 2016, <https://vefa.difi.no/bb/standard/asic/>
- [6] Dawn M. Turner, "ASIC - ASSOCIATED SIGNATURE CONTAINERS FOR EIDAS", CRYPTOMATHIC, April 2016, <https://www.cryptomathic.com/news-events/blog/asic-associated-signature-containers-for-eidas>
- [7] Kohsuke Kawaguchi, Windows service wrapper (winsw) documentation, 2016, <https://github.com/kohsuke/winsw>
- [8] Portal de administración electrónica, Gobierno de España, 'Formatos de Firma', 2017, <http://firmaelectronica.gob.es/Home/Ciudadanos/Formatos-Firma.html>
- [9] ETSI, 'ETSI TS 119 612: Electronic Signatures and Infrastructures (ESI); Trusted Lists', Version 2.1.1, July 2015.
- [10] Oracle, keytool and JKS documentation, 2017, <http://docs.oracle.com/javase/6/docs/technotes/tools/solaris/keytool.html>
- [11] Pivotal, 'Building REST services with Spring', 2017, <http://spring.io/guides/tutorials/bookmarks/>
- [12] John Moehrke, 'what is a Connectathon?', November 30, 2013, <https://healthcaresecrecyprivacy.blogspot.com.es/2013/11/what-is-connectathon.html>

Glosario

e-Sens	electronic Simple European Networked Services
SMP	Service Metadata Publisher
SML	Service Metadada Locator
ASiC	Associated Signature Container
UBL	Universal Business Language
REST	Representational State Transfer
MLR	Message Level Response
EoI	Expression of Interest
EoIC	Expression of Interest Confirmation
CfT	Call for Tenders
TSI	Tender Status Inquiry
SBD	Standard Business Document

Anexos

A Manual de instalación

Ejecución y depuración de esens-backend

Ambos proyectos cuentan con un ejecutable java, por lo que para ejecutarlos en ambitos de Desarrollo y pruebas simplemente se ejecutará las clases Java Application.

Instalando esens-backend como un servicio en Windows

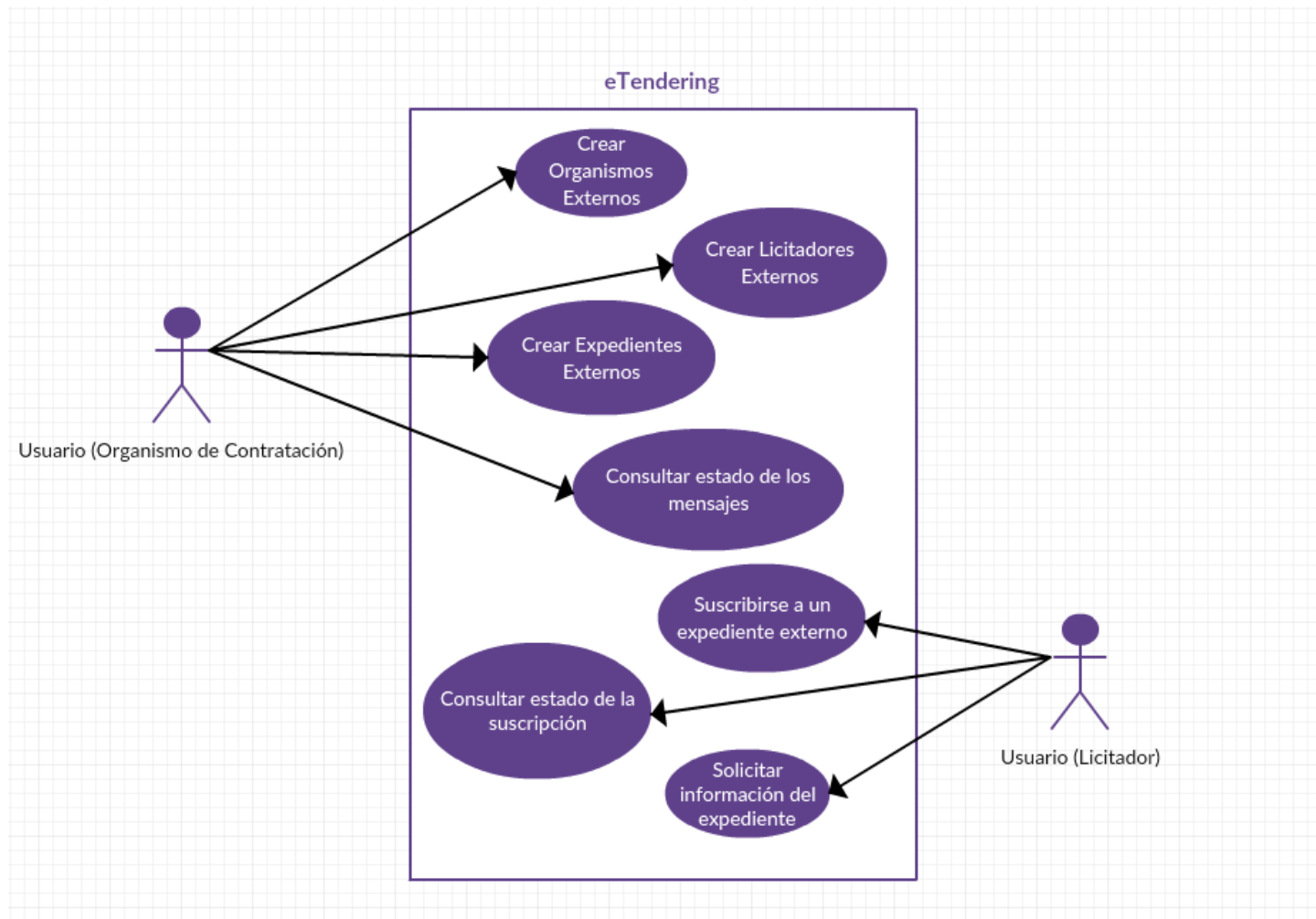
Para instalar el backend de eSens como un servicio de Windows nos ayudamos del ejecutable winsw, un wrapper de aplicaciones Java para crear este tipo de servicios.

En nuestro caso el comando mvn package sobre el proyecto maven creará el fichero zip donde se encontrará nuestro instalable. dentro de este fichero se encontrarán los siguientes elementos:

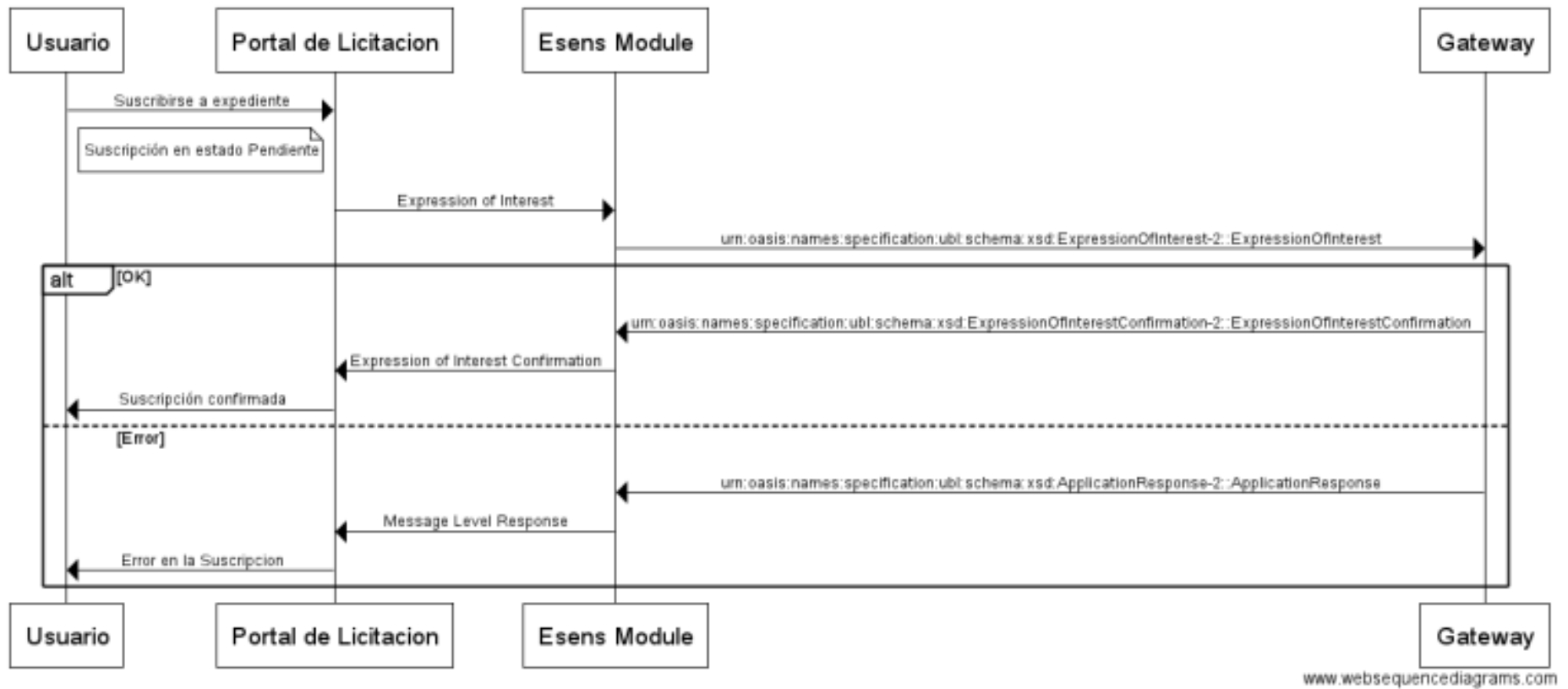
- esens-backend.exe: el ejecutable que se usará como servicio de Windows. Este ejecutable activará una máquina Java y la mantendrá activa. Para más información, realizar `esens-backend help` en cmd.
- fichero de configuración del servicio: dentro de estos podremos encontrar los distintos parametros que usará winsw para crear nuestro servicio. A priori, *en estos ficheros no sería necesario modificar nada*, ya que en ellos se dan referencias a las clases que realizarán las verdaderas funciones de nuestro servicio. Entre algunos parámetros de configuración de la aplicación podemos encontrar los datos de los servicios, los directorios dónde ser realizarán las operaciones y los datos necesarios para ejecutar la aplicación.
- /lib: En él se encontrarán tanto el jar del servicio eSens como todas las dependencias adicionales de este.
- /logs: directorio donde se generarán los archivos de log. Estos archivos serán logs tanto de nuestro servicio como de winsw, habiendo tanto un fichero exclusivo de logs para ambos y un fichero log común para los dos.
- /config: ficheros de configuración de nuestro servicio. De no existir, el servicio pondrá las variables configurables por defecto. Esto último *no es aconsejable* ya que, si existe otro servidor HTTP *pueden querer acceder al mismo puerto*.

Para la instalación se usarán los comandos del plugin wins, que instalará un servicio Windows con el nombre pasado por parámetro. Para manejar el servicio se podrá usar los comandos de winsw o bien la ventana de Servicios de Windows.

B Diagrama de Casos de Uso



C Diagrama de Secuencia sobre un Expression of Interest































D Anexo Pasos de Connectathon de octubre y su resultado






















2	4.2	Pixelware CA = 9920:Q3890349H	Expression of Interest Confirmation	TenderNed EO = 0106:30058019	Pixelware- Procurement project: PA/JGB/290115/01	ok
9	4.1	ETHICS EO = 0096:11534430-1	Expression of Interest	Pixelware CA = 9920:Q3890349H	Pixelware- Procurement project: PA/JGB/290115/01	ok
10	4.2	Pixelware CA = 9920:Q3890349H	Expression of Interest Confirmation	ETHICS EO = 0096:11534430-1	Pixelware- Procurement project: PA/JGB/290115/01	Pixelware was not able to process the message and so did not send an EoIC. Reason: data of the EO was missing. Action: Adjustment by ETHICS.
17	4.1	Vortal EO = 9946:510049222	Expression of Interest	Pixelware CA = 0106:27366538	Pixelware- Procurement project: PA/JGB/290115/01	ok
18	4.2	Pixelware CA = 9920:Q3890349H	Expression of Interest Confirmation	Vortal EO = 9946:510049222	Pixelware- Procurement project: PA/JGB/290115/01	ok
25	4.1	Pixelware EO =9920:92920000T	Expression of Interest	TenderNed CA = 0106:27366538	TenderNed - Procurement project: 174	ok
26	4.2	TenderNed CA = 0106:27366538	Expression of Interest Confirmation	Pixelware EO =9920:92920000T	TenderNed - Procurement project: 174	ok
27	4.1	Pixelware EO =9920:92920000T	Expression of Interest	ETHICS CA = 0096:10213231-1	ETHICS - Procurement project: /esens_0101_20150304s.nsf/	ok

28	4.2	ETHICS CA = 0096:10213231-1	Expression of Interest Confirmation	Pixelware EO =9920:92920000T	ETHICS - Procurement project: /esens_0101_20150304s.nsf/	ETHICS was not able to process the message and so did not send an EoIC. Reason: unknown Action: Adjustment by ETHICS.
29	4.1	Pixelware EO =9920:92920000T	Expression of Interest	eVergabe CA = 9930:DE122268496	eVergabe - Procurement project: 765a9937-16d5-4513-92b4- cc9800925951	ok
30	4.2	eVergabe CA = 9930:DE122268496	Expression of Interest Confirmation	Pixelware EO =9920:92920000T	eVergabe - Procurement project: 765a9937-16d5-4513-92b4- cc9800925951	ok
31	4.1	Pixelware EO =9920:92920000T	Expression of Interest	Vortal CA = 9946:507751639	Vortal - Procurement project: PT1.REQ.1	ok
32	4.2	Vortal CA = 9946:507751639	Expression of Interest Confirmation	Pixelware EO =9920:92920000T	Vortal - Procurement project: PT1.REQ.1	ok

E Capturas de la Connectathon de marzo

e90bbd1a-4ff4-400a-b152-05eb1cba5b88	 0106:30058019	 9920:Q3890349H	ExpressionOfInterest	30/03/2017	10:08	Correcto	Entrante
b9163b22-dc86-429c-a537-cb4ec3fa0606	 9920:Q3890349H	 0106:30058019	ExpressionOfInterestConfirmation	30/03/2017	10:09	Correcto	Saliente
d972843d-e784-4741-8097-6f1c47715689	 9946:510049222	 9920:Q3890349H	ExpressionOfInterest	30/03/2017	10:15	Correcto	Entrante
c8e8d658-4c15-42ab-9734-5508529188aa	 9920:Q3890349H	 9946:510049222	ExpressionOfInterestConfirmation	30/03/2017	10:16	Correcto	Saliente
03ad0c96-bbc4-4eec-9da6-39f1da1beece	 9920:92920000T	 0106:27366538	ExpressionOfInterest	30/03/2017	10:19	Correcto	Saliente
d5271476-c541-46f4-b512-ab1759039baf	 0106:27366538	 9920:92920000T	ExpressionOfInterestConfirmation	30/03/2017	10:20	Correcto	Entrante
905a9163-053c-4ba7-abb0-2b234ac0c819	 9920:92920000T	 9930:DE122268496	ExpressionOfInterest	30/03/2017	10:20	Correcto	Saliente
3ef5dfbc-04db-44e7-b740-76de7294ccc4	 9930:DE122268496	 9920:92920000T	ExpressionOfInterestConfirmation	30/03/2017	10:21	Correcto	Entrante
be3bd4e3-03da-471a-ab84-ecfbee904329	 9920:92920000T	 9946:507751639	ExpressionOfInterest	30/03/2017	10:21	Correcto	Saliente
761ba03c-acbc-4ccd-a330-40f050192c20	 9946:507751639	 9920:92920000T	ExpressionOfInterestConfirmation	30/03/2017	10:21	Correcto	Entrante

8619e5f3-88a9-424d-9e4f-9f29c4f040c5	 0106:30058019	 9920:Q3890349H	TenderStatusInquiry	30/03/2017	10:24	Correcto	Entrante
ae768952-4d16-4105-b132-0d8cf27c0154	 9920:Q3890349H	 0106:30058019	CallForTenders	30/03/2017	10:25	Correcto	Saliente
e19a89e9-889d-4779-8cb4-143beb129114	 9920:Q3890349H	 9946:510049222	CallForTenders	30/03/2017	10:29	Correcto	Saliente
46bc3ed6-3ee8-4c52-b485-3cab17a0eb66	 9946:510049222	 9920:Q3890349H	TenderStatusInquiry	30/03/2017	10:29	Correcto	Entrante
2777e157-ee70-4456-93da-e6a67e0ce8ac	 9920:92920000T	 0106:27366538	TenderStatusInquiry	30/03/2017	10:31	Correcto	Saliente
09a99c45-b0f2-4f0f-96ca-7ea2177dc799	 9920:92920000T	 9930:DE122268496	TenderStatusInquiry	30/03/2017	10:32	Correcto	Saliente
7cd56a01-a08f-4bff-88a9-89c38c9e7ba3	 9930:DE122268496	 9920:Q3890349H	CallForTenders	30/03/2017	10:32	Correcto	Entrante
7a95c554-7c47-4f74-b6cb-614411779907	 9920:92920000T	 9946:507751639	TenderStatusInquiry	30/03/2017	10:33	Correcto	Saliente
3a35d9b0-c25c-4c4c-9616-58022bdf4139	 9946:507751639	 9920:Q3890349H	CallForTenders	30/03/2017	10:33	Correcto	Entrante

ec579dcd-16ba-4a71-800c-5b672d48955e	 9920:92920000T	 9920:Q3890349H	ExpressionOfInterest	30/03/2017	11:17	Correcto	Saliente
3f9d0b7a-9473-49cc-b498-ca387660c50f	 9920:Q3890349H	 0106:30058019	ApplicationResponse	30/03/2017	11:17	Correcto	Saliente
b6ef8dbf-b8b7-40e0-b37f-d3e30850b86f	 9920:Q3890349H	 9920:Q3890349H	ApplicationResponse	30/03/2017	11:17	Correcto	Saliente
dc924ac9-8594-46e3-947c-e18899a596da	 0106:30058019	 9920:Q3890349H	ExpressionOfInterest	30/03/2017	11:17	Error de Validacion	Entrante
1b813c49-fe3b-499e-b1c1-69c1a0087d2c	 9920:Q3890349H	 9946:510049222	ApplicationResponse	30/03/2017	11:20	Correcto	Saliente
68198595-8a50-4b84-8807-db9639d59971	 9946:510049222	 9920:Q3890349H	ExpressionOfInterest	30/03/2017	11:20	Error de Validacion	Entrante
de418942-9daa-47a9-bcaa-b877ed8ae89b	 9920:92920000T	 0106:27366538	ExpressionOfInterest	30/03/2017	11:23	Error de Validacion	Saliente
18ccf4b9-66bb-42c3-af67-018ee676ff76	 9920:92920000T	 9930:DE122268496	ExpressionOfInterest	30/03/2017	11:23	Fallo de Validacion	Saliente
660f4647-79b3-4d1a-abc7-edd7f7495e97	 9930:DE122268496	 9920:Q3890349H	ApplicationResponse	30/03/2017	11:23	Correcto	Entrante
31fe1456-3235-463d-8bd4-d7dbed2d2192	 9920:92920000T	 9946:507751639	ExpressionOfInterest	30/03/2017	11:23	Error de Validacion	Saliente
1ac1860b-3f15-4303-9456-be4eb0ccb0c2	 0106:27366538	 9920:Q3890349H	ApplicationResponse	30/03/2017	11:23	Correcto	Entrante
2253810c-b56b-45cd-aefb-1f7dea0b4950	null:507751639	null:Q3890349H	ApplicationResponse	30/03/2017	11:23	Correcto	Entrante

